Norwegian School of Economics

Bergen, Autumn 2020

How do microeconomic determinants constitute the freight rate of individual contracts in the VLCC spot market?

Phuong Thi My Nguyen and Oanh Thi Hoang Pham

Supervisor: Roar Os Ådland

Master thesis, Economics and Business Administration

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Acknowledgements

We would like to express our sincere gratitude to our supervisor, Roar Os Ådland, for introducing us to the topic and providing invaluable guidance. His profound knowledge and experience in the shipping industry have been a motivation and great support for us. It was our great honor to conduct this thesis under his guidance. We hope that there will be other opportunities to work with him in the future.

This thesis cannot be completed without the whole-hearted support and caring from our families and friends. We are extremely grateful for the presence of those people in our lives.

Norwegian School of Economics

Bergen, December 2020

Oanh Thi Hoang Pham

Phuong Thi My Nguyen

Abstract

In this paper, we build a hedonic price model to explain the variation in freight rates in individual contracts using microeconomic data of the VLCC market. Using XGBoost and SHAP values to investigate the importance and significance of the various variables, we found that market condition and cyclicality have the greatest impact on the variance of freight rates, followed by route and charterer identity as a result of strategic behavior and bargaining power of charterers. Moreover, dominant charterers on westbound routes possess bargain power to reduce the fixture rates while their counterparts on eastbound routes are willing to pay a higher price than mean estimation. Finally, SHAP value can be considered as an efficient and reliable tool to assess the fixture rates in individual contracts.

Keywords – NHH, Master Thesis, Freight rate, Oil tankers, Generalized Additive Models (GAM), Extreme Gradient Boosting (XGBoost), SHapley Additive exPlanations (SHAP)

Contents

List of Figures

List of Tables

1. Introduction

The tanker shipping sector is one of the most active markets in the shipping industry with the highest trading volume. Crude oil tankers are mostly used to deliver crude oil from production points to the consumption area. Besides, they are used sometimes for storing post-produced crude oil and delivering oil products. The global crude tanker fleet size is forecasted to reach 423 million deadweight tonnes (Dwt) in 2020, a growth of 8.7% compared to 2018 (Research and Markets, 2020). The development of this market follows the increase of oil demand and urban population as long as geopolitical developments. Depending on the sizes of vessels, the tanker fleet is divided into five segments: VLCC (Very Large Crude Carrier), Suezmax, Aframax, Panamax, Handy and small tankers. Among these, VLCC tankers can transport up to 2 million barrels of oil and with a carrying capacity of around 300,000 Dwt and are considered to be more economical than other tankers, especially in transporting high volume of crude oil over long distances. Hence, "a charterer always prefers to hire a VLCC rather than chartering two or three Aframax" (Alderton, 2004). In 2019, the highest market share belongs to VLCC, followed by Suezmax and Aframax (Research and Markets, 2020). The freight market in the international bulk shipping industry can be mainly divided into spot market for single voyages and auxiliary market for period time charters (Adland, 2008). Under spot charter contracts, charterers hire the ship to carry specific cargo from a loading port to a discharge port and the price is specified as per-day rate or per-ton carry amount. On the other hand, time charter contracts are under a specific period of time, often at least a month, and the fixture rate is influenced by expectations about future short-term freight rates, interest rates, and risk premium (Kavussanos and Alizadeh, 2002). This paper will focus on the VLCC spot market.

The freight rate in the spot market can be determined by current supply and demand of the tanker shipping market (Stopford, 2009) or global economic activities. However, at the micro-level, the characteristics relating to vessels, route, and other contract specifications also play a role in forming fixture rates (see, for instance, Alizadeh and Talley, 2011a and Adland, 2016). This can be attributed to the shipping market practice that a fixture is often generated as a result of an auction among available vessels that are nearby the cargo.

With the development of artificial intelligent techniques recently, there is a rise in research using state-of-the-art models. Although the black-box and flexible nature of those models have been mostly employed to predict the future freight rate, it has not been useful for evaluating the

formation of freight rates. Balancing between the ability of capturing sophisticated patterns and interpretability is one of the most important considerations to derive desirable empirical results. Our paper attempts to model freight rates using advanced techniques yet offer an assessment of the contribution of microeconomic variables to individual fixtures. In particular, our study suggests an approach to explain the price formation for individual fixture rates in the VLCC market using microeconomic data from 2011 to 2020 obtained from Clarkson Research's Shipping Intelligence Network. Our contributions are twofold: (1) Building sophisticated models to estimate fixture rates using various microeconomic variables, and (2) providing an efficient approach to assess the rate estimations of individual contracts. The choice of microeconomic variables is greatly inspired by Adland, 2016. A statistical and semi-parametric model (i.e., GAM) and an artificial intelligent model (i.e., XGBoost) are formed to satisfy our first objective. The estimates from the model with better performance are used to measure SHAP values which in turn, reveal the contribution of each variable to individual contracts.

In the next chapter, we cover literature review of previous and current research about forecasting freight rates in the shipping industry. Chapter 3 develops an understanding of the data set, followed by the introduction and explanation of machine learning frameworks used in the study in chapter 4. Chapter 5 presents in detail our methodology to implement theoretical and empirical frameworks to analyze the data. Chapter 6 discusses the findings, while conclusion and limitations would be presented in chapter 7.

2. Literature Review

Investigating the formation of freight rate of bulk shipping has been a mature subject within maritime literature due to the availability of data and the maturity of the market. Based on the type of information used, the literature on this topic can be divided into two main groups.

The first one builds investigating models based on macro-level information which is mostly represented by the interaction of supply and demand and the published freight rate indices. Both continuous time models (e.g. Bjerksund and Ekern, 1995; Tvedt, 1997; Adland and Cullinane, 2006; Adland et al., 2008, Poblacion, 2015; Población, 2017) and time-series models (Kavussanos, 1996; Berg-Andreassen, 1996; Franses and Veenstra, 1997; Kavussanos and Alizadeh, 2001) are widely used in this approach. The performance of those studies has been poor probably because of aggregation bias (Alizadeh and Talley, 2011a).

The other group focuses on using micro information (i.e., specifications of individual vessels, routes, charterer, and owner, etc.,) as the input. Using a database of Worldscale fixtures over a period of four and a half years, Tamvakis (1995) forms several statistical tests to detect the presence of premium paid for vessels of lower age, double hull construction, or trading to the U.S.A. Tamvakis and Thanopoulou (2000) investigates the existence of a two-tier spot freight market in the dry bulk freight market for medium and large bulk carriers of differing ages. However, they found no statistically significant difference in rates paid among older and younger carriers. Laulajainen (2007) investigates differences in shipping freight rates and operational profitability for different routes. Alizadeh and Talley (2011a,b) concludes that the duration of the laycan period is an important determinant of the shipping freight rate, besides the vessel's hull type, fixture deadweight utilization ratio, vessel age, and voyage routes. Moreover, freight rates also have a great impact on the laycan period, besides the Baltic Dirty Tanker Index and its volatility. Using generalized additive models (GAMs), Köhn and Thanopoulou (2011) suggests that contract specifications (i.e., place of delivery, charter length and number of days forward to delivery, vessel size and consumption, the paper quantifies quality) are related to differences in physical dry bulk charter rates. Tvedt (2011) develops a theoretical framework to model short-run freight rate at the micro level of matching individual cargoes and vessels in the market for VLCCs out of the Persian/Arabian Gulf (AG). The paper suggests that market psychology plays an as important role as supply and demand in forming short-run freight rates and that there is a difference in bargain power among different matches

of charterer and owner, impacting the fixture rates. Agnolucci et al. (2014) investigates the existence of a time charter rate premium for fuel efficiency in the Panamax dry bulk market. Adland and Cullinane (2016) studies the contribution of charterer and owner to freight rate and concludes that time-invariant factors and market features are the most influential factors determining the spot freight rate variations for VLCC-Capsize markets. Furthermore, charterer fixed effect has a great impact on price in the VLCC market while charterer and match effect (i.e., the interaction between charterer and owner) are prominent contributors to the price in the Capsize market. Adland et al. (2017a) builds a model including macro, vessel, and contract-specific variables in order to find out the existence of fuel-efficiency premium in the dry bulk time charter market. They conclude that a premium is rewarded for energy-inefficient vessels during boom times and that later, owners recoup a small ratio of the savings in fuel costs through higher time charter rates. Adland et al. (2017b) estimates a hedonic pricing regression to produce a more objective market index based on heterogeneous fixture data in the Offshore Support Vessel (OSV) market. The paper concludes that the time fixed effects used to estimate the market index explain 70–80% of the variation in day rates and that spot freight rates are positively correlated with engine power and transport capacity. As a complement of Adland et al. (2017b), Adland (2019) uses transaction-based information to form a hedonic pricing framework to generate shipping indices and compare it to expert-generated price indices. They found a substantial deviation between their transaction-based indices and shipbrokers' market indices, which is positively impacted by the level of day rates, and negatively impacted by the volume of transactions.

Most of the mentioned literature use statistical and econometric based models. While offering interpretability, the functions of those models may not be flexible enough to capture fully complex patterns such as non-linearity, cyclicality, etc. In recent decades, artificial intelligent techniques with their flexible function designs and powerful self-learning capabilities to produce more accurate results are becoming more popular as an alternative approach. However, artificial intelligent techniques are also regarded as "black-box" approaches as there is almost no transparency in how they treat the input information to generate the outcomes, a challenge in cases that the users desire to obtain more insight from the models in order to draw informed decisions. The attention about the trade-off between model accuracy and its interpretability has been rising in recent times. There have been several proposed approaches to address this problem such as LIME, DeepLIFT, Layer-wise Relevance Propagation, etc. Lundberg and Lee

(2017) suggests a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations) which assigns each feature an importance value for a particular prediction. This idea evolved from the concept of “Shapley values” in game theory for cooperation games (Shapley, 1953).

The contribution of our paper is to exploit the flexible nature of advanced models to better capture the non-linear and cyclical patterns of the tanker shipping market yet maintain the explanatory ability of the hedonic price model using microeconomic determinants for the VLCC market. Therefore, we conduct a statistical and semi-parametric model (i.e., GAM) and an artificial intelligent model (i.e., XGBoost) to model the freight rates. The estimates from the model with better performance are used to derive SHAP values to evaluate the importance and significance of various microeconomic variables on the formation of fixture rates. SHAP values is also an efficient tool to assess fixture rates of individual contracts.

3. Data

This chapter aims to discuss the dataset that is employed to estimate the models. After collecting the relevant dataset, we process the necessary steps to gather the set of variables for the models. As soon as the pre-processing of data is finished, the overview of each variable as well as the relationship between variables are given.

3.1. Data Collection

The dataset is derived from Clarkson Research's Shipping Intelligence Network (2020) and includes 16,495 observations for the VLCC spot market from 4th January 2011 to 17th September 2020. The original data provides information of fixture *dates*; *laycan from* and *laycan to* dates which are the earliest day and the latest day that vessel has arrived at the port of loading and is ready to be loaded; names of *charterers* and *owners*; information of *loading* and *discharge* ports; and other information of vessels such as years when vessels were *built*, *deadweights* (Dwt), transported *quantities*; *freight rates* which are the dependent variable of the study as well as the *unit* of freight rates, namely WS (world scale), USD or RNR (rate not reported)[1].

World scale is developed by the World Scale Association in London as an attempt to return the same net daily income irrespective of voyage performed. WS100 is the flat rate which represents the voyage costs (expressed in USD per metric ton of cargo) of a standard vessel[2] by transporting a tonne of cargo on an average 15,000-mile round trip voyage (Worldscale Association Limited, n.d.). The flat rates are set annually based on the distance, a standard vessel's fuel consumption, an average speed, updated vessel bunker prices, transit fees and the port costs and exchange rates (Stopford, 2009). The freight rate is negotiated upon the percentage of the flat rate WS100. Hence, WS50 means the price is one half of the published flat rate.

[1] Clarkson Research claimed that the unreported charter rates might happen when the various broking houses/Baltic Exchange reported the same fixture. However, unavailable freight rates are mainly for confidential reasons (Parker, 2014).

[2] A standard vessel is defined as having 75,000 Dwt, consumes 55 tonnes of fuel oil per day while sailing with additional 100 tonnes for other purposes than steaming and 5 tonnes of fuel in port (Stopford, 2009).

3.2. Data Pre-processing

From the original dataset, we select variables relating to vessel, route, and fixture to explain the variation in freight rates in the VLCC spot market. Our choice of microeconomic determinants very much follows what is indicated in literature sections, especially the set of variables proposed by Adland et al. (2016) with some adjustments.

Both Alizadeh and Talley (2011a) and Adland et al. (2016) include market index, which relies on Baltic Index, as a macro-determinant in the formation of freight rates. However, Adland et al. (2017b) argues that using the brokers' market indices may generate biased estimations. Firstly, the indices may contain part of the heterogeneity that is already accounted for in the set of explanatory variables. Moreover, transaction-based data is greatly impacted by a macro variable, the market index, which is derived a priori from the micro data itself, resulting in a circularity problem. Therefore, in our study, the market index is generated following the procedure introduced by Adland et al. (2017b) and Adland et al. (2019). In particular, the transaction *date* is used as a time-series indicator of the market. Adland et al. (2017b) also sounds a note of caution concerning the choice of time unit which should be long enough so that none of the time buckets are empty (i.e. does not contain any fixtures) as in this case, the estimate of the market level is zero. Thus, we choose to present the market indices at a monthly level.

Lead time is measured as the duration between *laycan from* date and transaction date. *Lay time* thus refers to the agreed period of time the vessels are allowed to load or discharge and is suggested by Alizadeh and Talley (2011a). Variable *age* is the vessel age when the fixture transaction occurred and is calculated based on the year when the vessel was *built*. *Load factor* is the utilization ratio between the transported *quantity* to the total vessel's freight capacity or *deadweight.* The final explanatory variable is the *route* from *loading* ports to *discharge* ports. There are 21 *routes* in total, represents the most popular routes, including Persian Gulf - South Korea, Persian Gulf - China, Persian Gulf - East, Persian Gulf - West, Persian Gulf - India, Persian Gulf - USG, Persian Gulf - Taiwan, Persian Gulf - Japan, Persian Gulf - Singapore, Persian Gulf - Malaysia, Persian Gulf - South Africa, Persian Gulf - Thailand, Persian Gulf - Canada, Persian Gulf - UKC, West Africa - USG, West Africa - China, West Africa - Taiwan, West Africa - East, West Africa - UKC, West Africa - India; and Other (i.e., routes in which

less than 20 transactions took place). Finally, we have a list of 8 explanatory variables: *market index, charterer, owner, lead time, vessel age, load factor, route,* and *freight rate* is the dependent variable.

From the list of 16,495 observations, we exclude transactions that do not record names of *charterers* and *owners*, *quantity, deadweight*, or *built*. Those observations account for 50.6% of the number of transactions. Furthermore, only fixtures with freight rates that are given in terms of the Worldscale index (i.e., an attempt at normalizing an implied USD/tonne rate across all routes) are selected. The choice of the Worldscale index simplifies the comparison of market levels for different vessel sizes and trade routes[3]. There are 57.5% of observations that meet this condition.

Finally, we exclude observations with negative values of *lead time*, which implies the delayed reporting of fixtures. Observations with *lead time* more than 50 days and *vessel utilization ratio* more than 1 or less than 0.7, which probably due to wrong input, are also omitted. A small minority of observations (1.53%) suffered from *lead time* and *load factor* constraints, leaving us with 7,485 observations.

Figure 1 gives a bird's eye view of the response variable. The VLCC tanker market experienced considerably stable freight rates during 2011-2018 but skyrocketed at the last two years of the observation period (2019-2020). Subsequent step is to detect outliers of the response variable.

[3] The flat rate WS100 is reviewed annually for all routes. Hence charter rates are not completely comparable across calendar years and for large vessels due to changes in bunker prices, pilotage fees, or other associated costs.

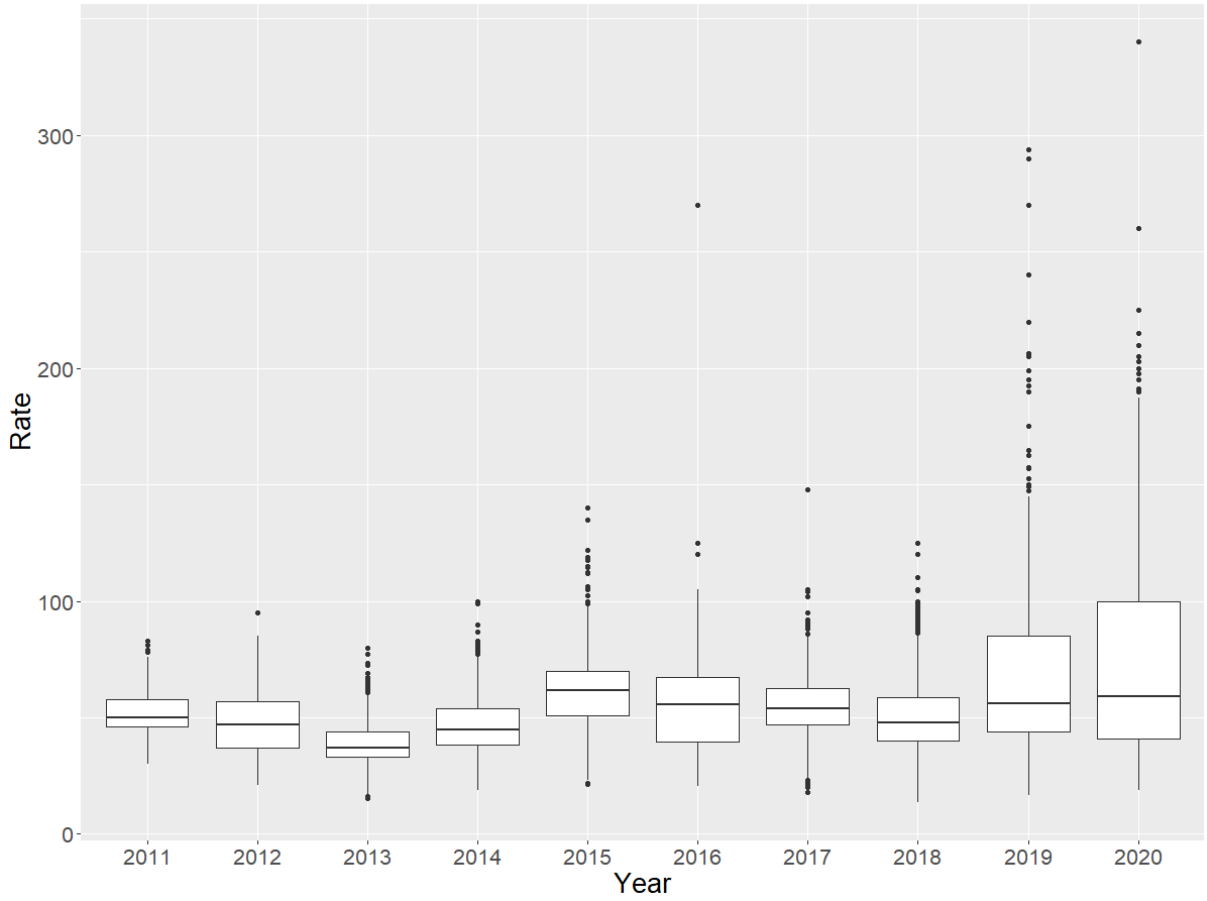


Figure 1 - Overview of annual freight rates. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

As can be seen from figure 2A, the freight rates follow the right-skewed distribution. It is not obvious whether the data contains meaningless outliers that may decrease the statistical power of the model at later stages since those extreme cases possibly contain noteworthy information. Therefore, instead of removing outliers, we implement log-transformation on the *freight rate* variable. Log-transformation is also advisable to handle outliers when the response variable follows the right-skewed distribution. There is a considerable number of recent studies dedicated to log-transformation in an attempt to stabilize the variance of prices such as Alizadeh and Talley (2011a), Adland et al. (2016), and Adland et al. (2017a). The distribution following log-transformation is still slightly skewed because those outliers are widely scattered. However, the transformation converts the original distribution closer to the normal distribution (figure 2B).

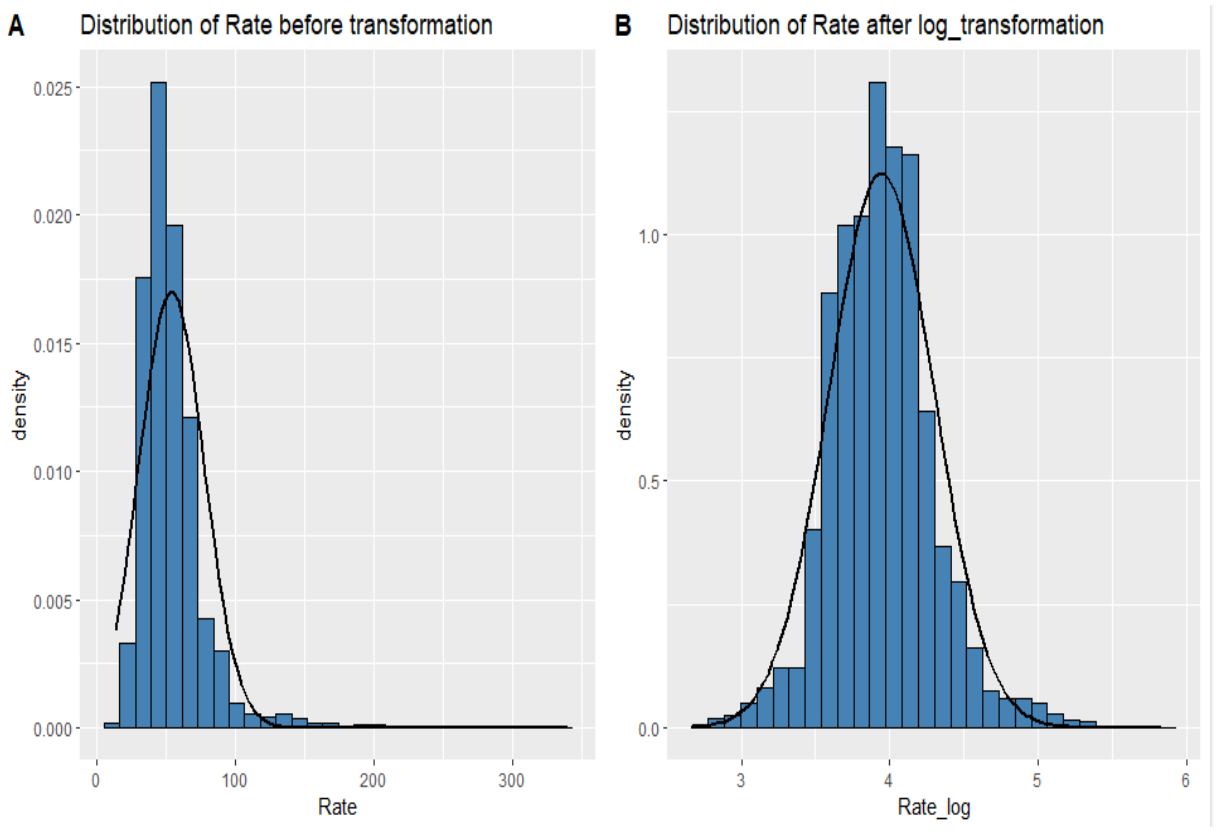


Figure 2 - Distribution of freight rates before and after log-transformation. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

It is worth noting that the numeric variables of the original data are different in units and range. Although rescaling is widely considered to conduct multivariate analysis variables with comparable units, we do not implement it as the magnitude of variables does not impact the decision tree-based model, which will be generated at a later step[4].

3.3. Descriptive Statistics

Table 1 provides an overview of all numeric variables. The study observes *freight rate* after log-transformation ranging from 2.67 (rate is WS13.5 in 5th Mar 2018) to 5.83 (rate is WS340 in 16th Mar 2020) with the mean is 3.95 over the course of the observed decade. The mean vessel *utilization ratio* is 0.88 and ranges from 0.72 to 1.00. *Lead time* variable has an average

[4] Standardization is still considered for fixed effect regression models (Appendix A6) since linear regression is more sensitive to the magnitude of variables. This technique will ensure that mean and standard deviation of all numeric variables are 0 and 1, respectively.

of 17 days and varies from 0 to 50 days. The collected data of vessel *age* ranges from 0 to 23 years and the average age of vessels is 8.84 years. Although the expected life of a VLCC vessel is approximately 20 years, most charterers are reluctant to carry oil in old vessels (Euronav, 2017) as there would be higher risk from deterioration of the hull and lower fuel efficiency (Shipbroker, 2011). Evidently, only 66 fixtures are associated with the vessels which reach their 20th anniversary.[5]

Variables	Mean	Std. dev.	Min	25%	Median	75%	Max
Dependent variable							
Freight rate (log)	3.95	0.36	2.67	3.71	3.93	4.15	5.83
Quantitative explanatory variables							
Lead time (in days)	17.35	6.43	-	14.00	17.00	20.00	50.00
Vessel age (in years)	8.84	5.54	-	4.00	8.00	14.00	23.00
Utilization ratio (in %)	0.88	0.04	0.72	0.85	0.88	0.90	1.00

Table 1 - Descriptive statistics of numeric variables. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

Further analysis is carried out with three categorical variables: charterer, owner, and route. The top ten *charterers* and *owners* are identified in table 2. The top 10 charterers account for 61% of all fixtures, while the number for owners is 26.2%. A majority of shipowners demand higher prices than the average of WS54.3 (except Maran Tankers Mngt.), and the highest average price is set by Dynacom Tankers Mgmt (WS69.4). Meanwhile, four out of ten charterers agree higher than average charter rates.

Figure 3 provides insight into the frequency of *charterers*, *owners* as well as the interaction between pairs of *charterers* and *owners*. The two highest frequencies are between UNIPEC and Ocean Tankers with 77 transactions, and between IOC and New Shipping with 72 transactions. UNIPEC and IOC are the two world's largest charterers of oil tankers, while New Shipping is also one of the most active shipowners.

[5] Please refer to appendix A1 for a more detailed overview of the quantitative variables.

Charterers	Fixtures	Rate (mean)	Percentage	Cumulation	Owners	Fixtures	Rate (mean)	Percentage	Cumulation
UNIPEC	1340	57.5	17.9	17.9	Euronav NV	239	54.6	3.2	3.2
IOC	509	65.9	6.8	24.7	New Shipping	239	60.9	3.2	6.4
CHEVTEX	475	49.9	6.3	31.0	Maran Tankers Mgmt	224	61.5	3.0	9.4
S.OIL	415	47.9	5.5	36.5	DHT Management	217	56.1	2.9	12.3
DAY HARVEST	378	58.3	5.1	41.6	Aeolos Management	196	57.9	2.6	14.9
PTT	354	54.0	4.7	46.3	Maran Tankers Mngt.	193	47.7	2.6	17.5
SHELL	326	49.1	4.4	50.7	Dynacom Tankers Mgmt	173	69.4	2.3	19.8
EXXONMOBIL	280	39.8	3.7	54.4	Dynacom Tankers Mngt	165	54.8	2.2	22.0
RELIANCE	271	59.0	3.6	58.0	Ocean Tankers	160	60.2	2.1	24.1
CPC	252	51.3	3.4	61.4	Shpg Corp of India	148	58.0	2.0	26.1
Others	2885	53.8	38.5	100	Others	5531	53.0	73.9	100
Total	7485	54.3	100		Total	7485	54.3	100	

Table 2 - Top ten charterers and owners. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

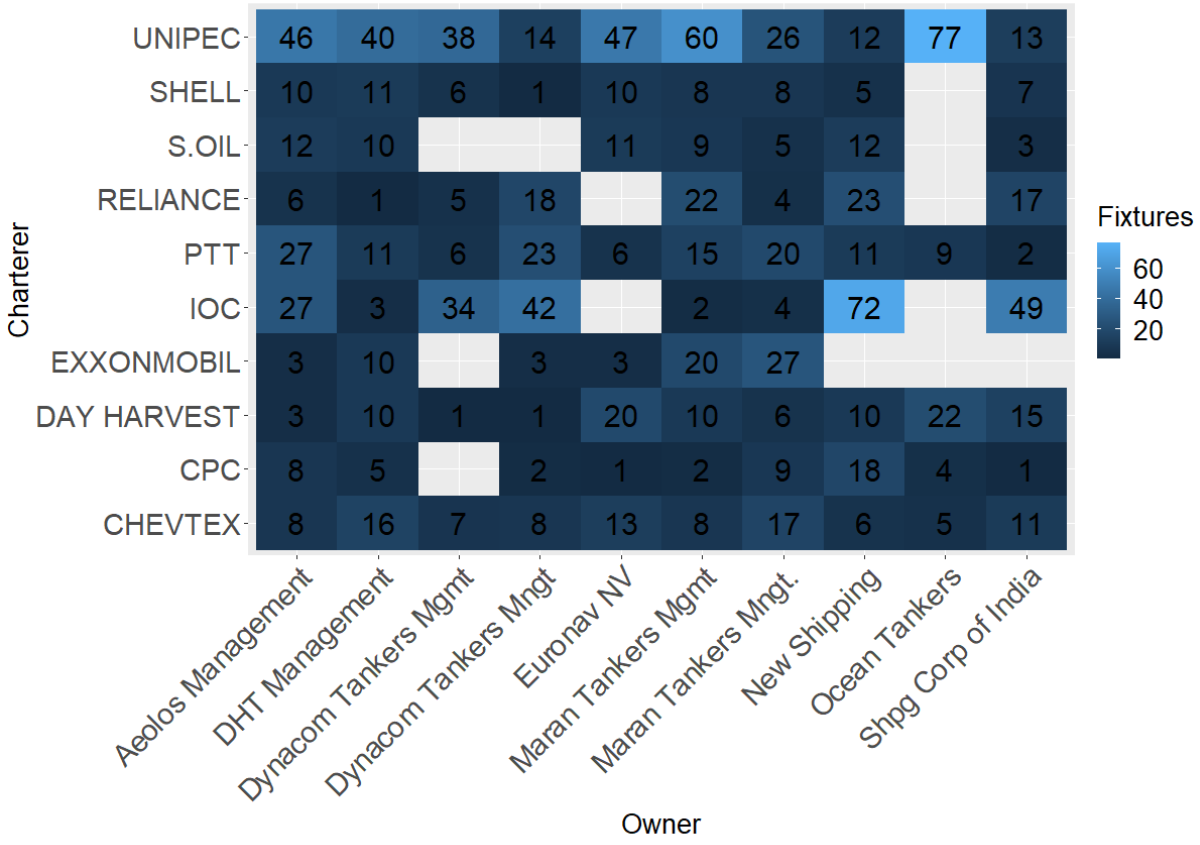


Figure 3 - Heatmap with top ten owners and charterers. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

Figure 4 below reveals the rank of routes according to their mean rates. In fact, the main loading area over the course of the decade is Persian Gulf, accounting for 84.1% of total fixtures. Persian Gulf - West, Persian Gulf - USG, Persian Gulf - UKC, and Persian Gulf - Canada have the

lowest logarithm of freight rates and are all westbound flows started from Persian Gulf as opposed to the higher prices in eastbound. This can be explained by the fact that shipowners discount freight rates of westbound trading routes in an attempt to obtain a backhaul in West Africa, while in contrast, eastbound routes need to ballast back to Persian Gulf (Parker, 2014).

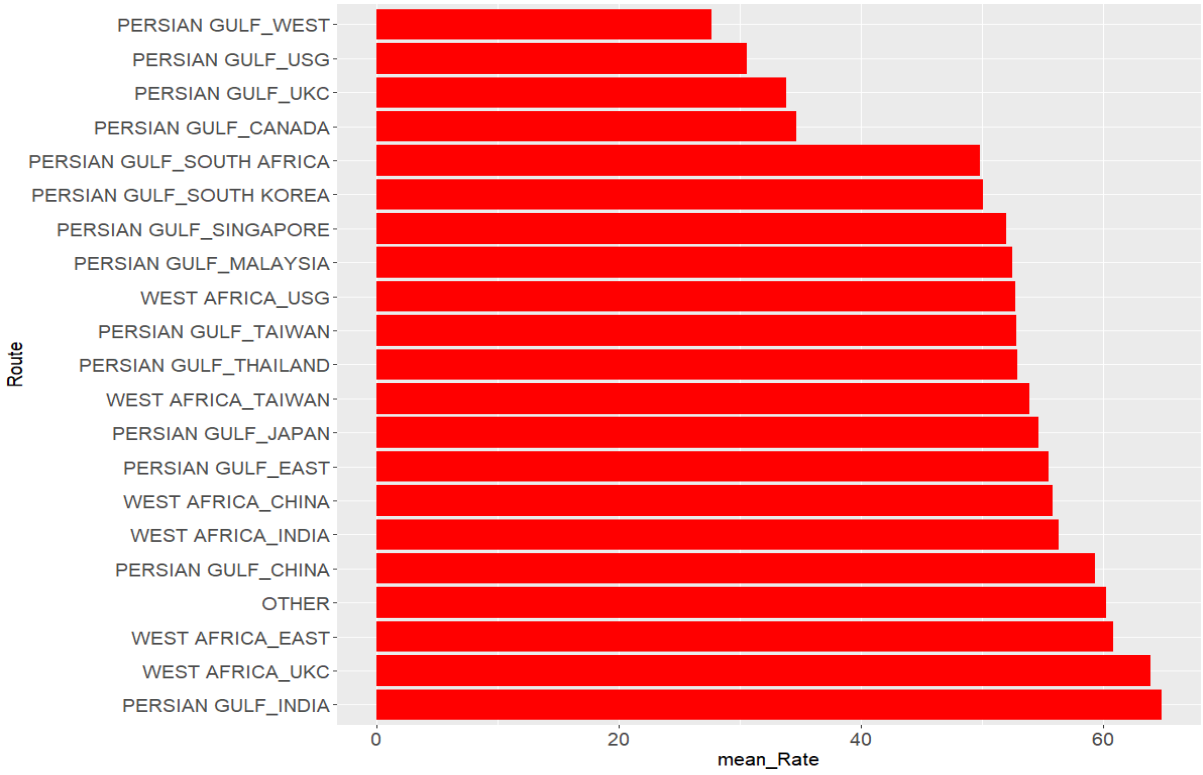


Figure 4 - Mean of freight rate of each route. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

To further analyze the top ten routes with the highest number of transactions, the mean values of related variables and their frequencies over time are presented in table 3 and figure 5, respectively. The most active routes mainly start from Persian Gulf to South and East Asia and account for 90% of total fixture transactions. Half of the list has mean rates higher than the average of all transactions (WS54.3). Most of the routes are associated with less than 10 years in average vessel's age, except for flows starting from Persian Gulf to India, Thailand, and Taiwan. Persian Gulf - USG has the lowest mean rate (WS30.63) but the highest mean utilization ratio (0.9%) and vessel age (6.12 years). In contrast, Persian Gulf - India possesses

the highest mean rate (WS64.88) and average vessel's age (12.58 years)[6]. Figure 5 shows the relatively similar trends among the top ten routes over time.

Route	Fixture	Freight rate (WS)	Utilization ratio (%)	Vessel age (year)	Lead time (date)
PERSIAN GULF_CHINA	1503	59.32	0.88	8.35	16.39
PERSIAN GULF_SOUTH KOREA	1243	50.08	0.88	8.84	16.25
PERSIAN GULF_INDIA	984	64.88	0.88	12.58	16.33
WEST AFRICA_CHINA	812	55.87	0.84	6.78	21.86
PERSIAN GULF_THAILAND	406	52.96	0.88	10.15	15.14
PERSIAN GULF_TAIWAN	396	52.87	0.88	11.37	16.14
PERSIAN GULF_USG	396	30.63	0.90	6.12	18.77
PERSIAN GULF_EAST	348	55.53	0.88	9.31	16.01
PERSIAN GULF_JAPAN	341	54.70	0.88	7.97	16.41
PERSIAN GULF_SINGAPORE	315	52.00	0.87	8.57	16.07
Others	741	49.76	0.87	6.80	20.28
All	7485	54.30	0.88	8.84	17.35

Table 3- Top ten routes and related statistics. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

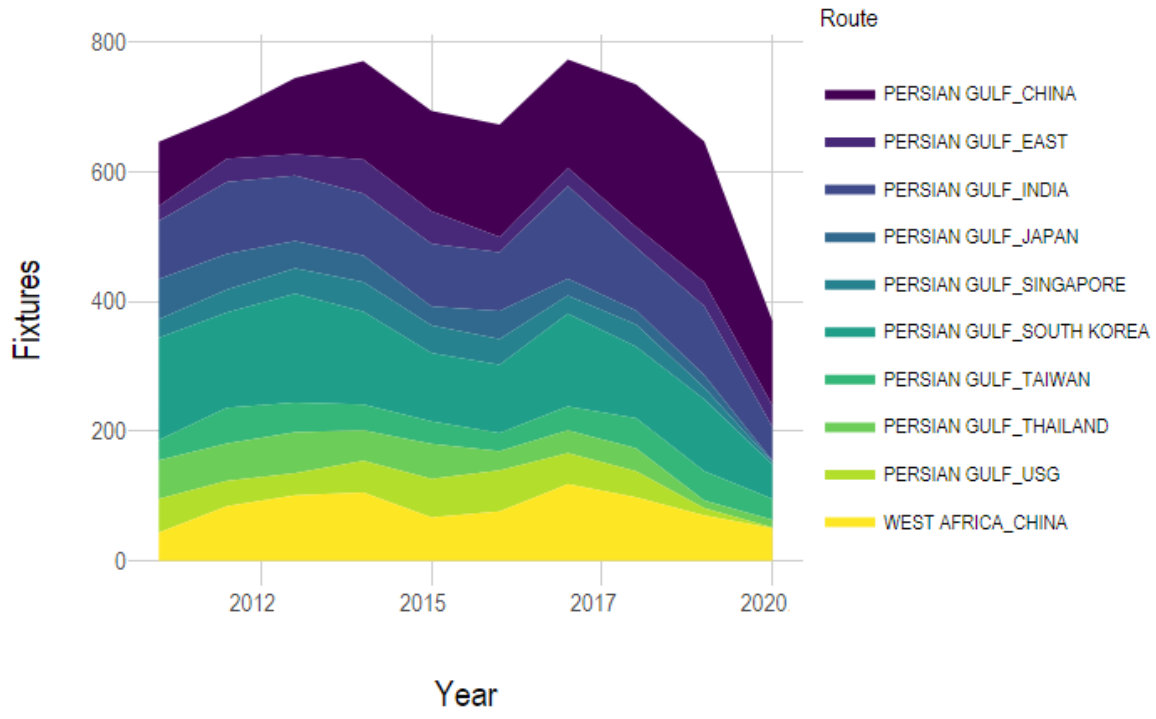


Figure 5 - Frequency of top ten routes. Source: Authors' calculations, data from Clarkson Research (January 2011–September 2020).

[6] Persian Gulf-India is also the shortest route among the top ten routes as opposed to Persian Gulf-USG, which has the longest distance. Detailed information of route distance is provided in appendix table A2.1

4. Machine Learning Theory

In order to explain the variation of freight rates in the VLCC oil tanker market, a statistical and semi-parametric model (i.e., GAM) and an artificial intelligent model (i.e., XGBoost) are formed. SHAP values facilitate the verification of the impact of each variable on individual contracts. This chapter addresses the underlying machine learning theory behind price models which will be analyzed in later sections: GAM, XGBoost, and SHAP, as well as the reasons behind the approach[7].

4.1. GAM

GAM (Generalized Additive Models) was first invented by Trevor Hastie and Robert Tibshirani in 1986. It is the extension of GLM (Generalized Linear Models) by assuming that the response variable is a sum of arbitrary functions of each dependent variable (Wood (2006b). Mathematically, GAM model has the structure as below:

$$g(E(Y_i)) = \boldsymbol{X}_i^* \theta + s(x_{1i}) + s(x_{2i}) + s(x_{3i}, x_{4i}) + \ldots \qquad (1)$$

where Y_i is the response variable and $E(Y_i)$ depicts the expected value of Y_i. $\boldsymbol{X}_i^*$ refers to a vector of any components that enter the model parametrically with a corresponding parameter vector θ. The acronym $s(x_i)$ denotes a smooth, nonparametric function of each dependent variable. Nonparametric means that the shape of variable functions is distribution-free or with unspecified parameters and purely determined by data.

GAM is selected due to its flexibility, interpretability, and regularization.

- Flexibility: GAM relaxes the linearity assumption by allowing each explanatory variable to have a non-linear pattern. However, instead of manually fitting many different parametric regression models and selecting the best models for each determinant, functions are automatically derived. Hence, GAM can capture the non-

[7] To verify the explanatory powers of more advanced methods, we need to compare our performance of the chosen models with one benchmark model. Linear regression models with charterer and owner fixed effect, time fixed effect are respectively formulated on the full dataset (Appendix A6). GAM and XGBoost models (Appendix A7 & A8) are generated on the full dataset as well to provide a concrete comparison between the benchmark model and more advanced methods.

linear relationships that linear models might miss in a time-consuming way (James et al., 2013).

- Interpretability: Since GAM remains the additive nature of linear regression, it holds interpretability advantage. Simply put, GAM allows us to examine the effect of each independent variable on the response variable while holding other variables fixed (James et al., 2013).
- Regularization: We can control the model's smoothness by adding a "wiggliness" penalty while fixing the basis dimension at a size slightly larger than reasonably necessary. In other words, instead of fitting the model by minimizing ||y - Xβ||, it could minimize:

$$\left|\left|y - X\beta\right|\right|^2 + \lambda \int_0^1 [s''(x)]^2 \, d(x) \qquad (2)$$

The trade-off between model's smoothness and goodness of fit is controlled by the smoothing parameter λ. The curve of data becomes smoother when increasing the value of λ. $\lambda = \infty$ denotes a straight-line estimate while $\lambda = 0$ denotes un-penalized estimate. Therefore, the too low or too high values of λ may lead to under smoothed or over smoothed curves. However, we can control λ by running restricted maximum likelihood (REML)8 (Wood, 2006b) in which the smooth is treated as a random effect.

4.2. XGBoost

XGBoost (Extreme Gradient Boosting) was created by Tianqi Chen and Carlos Guestrin in 2014 as an implementation of a gradient boosting framework with regularization factors. XGBoost immediately caught the attention. In recent years, XGBoost framework is dominating many machine learning competitions due to its outstanding speed and performance. Although XGBoost was based on the gradient boosting framework, it proved to be more than 10 times faster and more efficient by including two solvers: linear model and tree learning algorithms.

[8] The other common way of controlling λ is generalized cross-validation (GCV). However, in the course of our study, we conduct REML to select optimal λ since this approach tends to be more robust to under-smoothing than GCV (Wood, 2006b). REML score is computed automatically under package *mgcv* in R.

XGBoost uses objective function (loss function and regularization) to build trees by minimizing this equation:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}^{t-1} + f_t(x_i)) + \Omega(f_t) \quad (3)$$

$$\text{where } \Omega(f_t) = \gamma T + \frac{1}{2}\lambda||\psi||^2 \quad (4)$$

The first part of equation (3) is the loss function which is the difference between the fitted and the actual data. XGBoost uses Second Order Taylor Approximation for both regression and classification.

$$g_i = \partial_{\hat{y}^{t-1}} l(y_i, \hat{y}^{t-1}) \quad (5)$$

$$h_i = \partial^2{}_{\hat{y}^{t-1}} l(y_i, \hat{y}^{t-1}) \quad (6)$$

are the first and second derivative of the loss function, respectively. Then the loss function in model (3) can be simplified as:

$$L \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{t-1}) + g_i \times f_t(x_i) + \frac{1}{2} h_i \times f_t^2(x_i)] \quad (7)$$

The second part consists of the regularization term ɣ which penalizes T, the number of terminal nodes, or leaves in the tree to encourage pruning. The pruning part takes place as soon as the tree is built and does not impact similarity scores or optimal output values. w is the weights of different leaves and be scaled by the L2 regularization term λ, which is similar to ridge regression. The more emphasis we give the regularization penalty by increasing lambda, the closer the optimal value will get to 0.

There are some of the key features of XGBoost which contribute to the success of this machine learning method:

- Overfitting: One of the main risks of prediction is overfitting which is a situation when the model tries to capture as much noise of the training data as possible, leading to low accuracy on test data. Regularization adds additional penalty terms to errors and shrinks the coefficient of variables toward zero. By doing so, regularization can prevent the risk of overfitting (Morde, 2019).

- Missing Value: XGBoost can handle missing value issues by figuring out the trend of missing value and deciding the optimal direction of the nodes to go next in an effort to minimize loss (Dwivedi, 2020).
- Flexibility: XGBoost offers a wide range of applications, namely regression, classification solver, ranking, and even prediction defined by users (Dwivedi, 2020).
- Multicollinearity: XGBoost or boosting in general is more robust with multicollinearity compared to OLS regression. When two variables are highly correlated, it may be an issue to calculate OLS regression as the redundant features will impact the stability of the model. However, gradient boosting assesses the importance of features and leaves out the redundant features when it builds the tree.
- Feature importance and feature selection: This is one of the most attractive applications of XGBoost. It scores the contribution of all features in making key decisions to build the tree and ranks the importance scores. A more detailed description of this function will be introduced later.

4.3. SHAP (SHapley Additive exPlanations)

4.3.1. Shapley Value

Shapley value was first introduced in a celebrated 1953 paper by Lloyd Shapley, "A value for n-person games". There he suggested a method to measure numerically the contribution of each player in a cooperative game based on game theory. The application of Shapley value has evolved into numerous domains, one of which is machine learning. The Shapley value of a feature value is the average marginal contribution of its value across all possible coalitions. Intuitively, the process of measuring Shapley value of a feature value can be described as following: a feature value enters a room that already contains a set of features with their values. All the features' values in the room would contribute to the coalition game together. The Shapley value of a feature value is the average adjustment in the prediction when the feature values join the room compared to the mean prediction obtained from the current set of features' values (Molnar, 2019).

The Shapley value is a value function v of players in S with S is a subset of features. $v(S)$can be interpreted as the total expected sum of payoffs that the set of features in S can produce by

cooperation. The amount that feature i contributes given the subset *S*, *p* - the number of features in *S* is as subsequent:

$$\varphi_i(v) = \sum_{S \subseteq \{x_1,\dots,x_p\}\setminus\{x_i\}} \frac{|S|!\,(p-|S|-1)!}{p!}\left(v(S \cup \{x_i\}) - v(S)\right) \quad (8)$$

There are four properties of Shapley value that make it a fair approach to distribute payouts across features:

- Efficiency: The feature values must contribute to the difference between the prediction *x* and the mean value.
- Symmetry: The contribution of feature values *i* and *j* should be the same if they equally contribute to all possible subsets of features.
- Dummy: If the feature value *i* does not contribute to the prediction *x*, the Shapley value of *i* should equal 0.
- Additivity: In a random forest model, for instance, the Shapley value for a feature value for the random forest would be the sum of average Shapley values of each individual tree.

4.3.2. SHAP (SHapley Additive exPlanations) and TreeSHAP

SHAP (SHapley Additive exPlanations), which is introduced in Lundberg and Lee (2016), is developed from the concept of Shapley value in order to measure the prediction for an individual instance by measuring the contribution of each feature to the prediction. Lundberg and Lee (2016) proposed two SHAP approaches: KernelSHAP which is a Kernel-based estimation and TreeSHAP which is an efficient approach for tree-based models. Since in this paper, we will use the XGBoost model, we will only focus on TreeSHAP.

There are three important properties of SHAP:

- Local accuracy: which is equivalent to the property of Efficiency of Shapley value.
- Missingness: the feature that does not contribute to the prediction gets the attribute of 0.
- Consistency: this property refers to the change of feature value's contribution according to the change of model. This desirable characteristic of SHAP makes us choose it over XGBoost Importance.

Instead of marginal expectation, TreeSHAP uses conditional expectation to draw the value function. The conditional expectation is as subsequent:

$$E_{X_s|\ X_C}(f(x)|x_S) \qquad (9)$$

Although TreeSHAP is faster than each counterpart, KernelSHAP, it has a problem with using conditional expectation which is that the feature may get a non-zero TreeSHAP value even when it has no contribution to the prediction. The cause of this issue is the correlation of that feature with another one that has an impact on the prediction function.

5. Methodology

In this paper, we propose a methodology comprising three stages. First, we split the data into train-test sets, and encode categorical variables into numerical ones. Then, we formulate models using GAM and XGBoost framework. Finally, a more in-depth analysis of feature importance according to XGBoost and SHAP are presented.

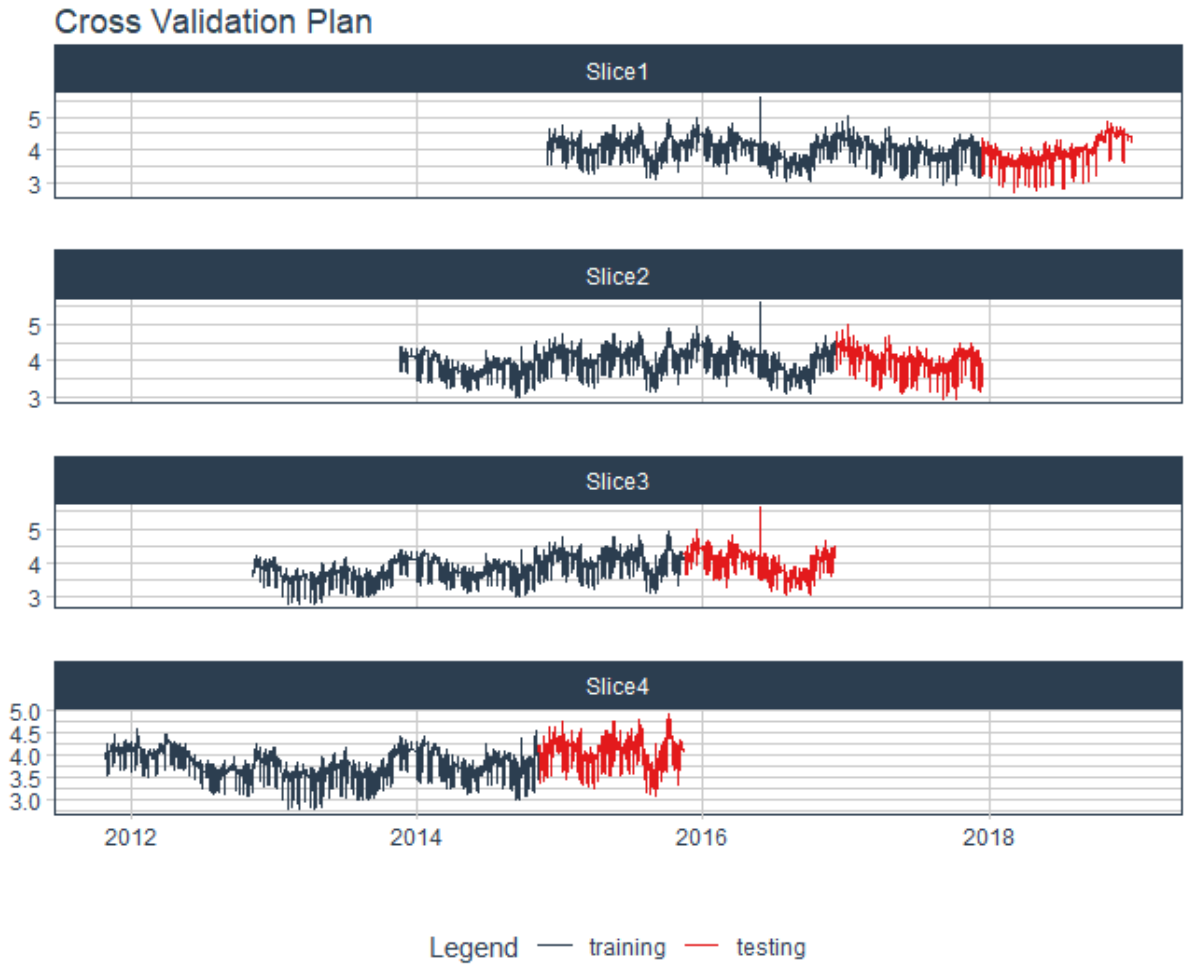
5.1. Preparation before Modeling

5.1.1. Train-Test Split

In an attempt to avoid overfitting and ensure the randomness of the dataset, random sampling and cross-validation are among the most common practices. In our study, these techniques are not appropriate as there might be a risk of future-lookingness when we train models. Rather, time-based splitting and blocked cross-validation enable us to fit and evaluate the training models while keeping temporal order which is a critical characteristic of time-series data. We subset the data into two chronological parts: training set including observations from 2011 to 2018, and a test set containing the last two-year observations[9]. The split ratio is treated with the utmost caution to balance a trade-off between estimated parameters and performance statistics. More specifically, if the training data is not sufficiently large enough, it may lead to higher variance in parameter, while much emphasis on the training set might make an unreliable estimation of model performance.

Under the XGBoost framework, we conduct blocked cross-validation in tuning hyperparameters to split the training set further into 4 slices. Each slice consists of observations in four years in which three years are used to train the models and the next one year is performed as the validation set. The outlook of the cross-validation plan is produced in figure 6.

[9] There are 6,367 observations in the training set and 1,118 observations in the test set. It is equivalent to a split ratio of 85:15.



Note: The value presented in the figure is logarithm of freight rate.

Figure 6 – Cross-Validation Plan

With the completion of the blocked cross-validation, we then compare performances of all models and select the best model with its optimal parameters. Finally, the test set is adopted to evaluate the model performance by the mean of accuracy measures. On the other hand, only the original train and test sets are adopted for the GAM model.

5.1.2. Target Encoding

As three of our independent variables (i.e., route, charterer, and owner) are categorical variables and XGBoost only deals with numeric variables, it is necessary to encode those variables properly to enable fitting of XGBoost model while maintaining information from the original data. We consider between one-hot encoding, which can translate categorical variables into

matrices of dummy variables, and target encoding, which is a Baysian encoding technique. However, the efficiency of one-hot encoding decreases significantly if there is a large number of levels present in the data. In our data, there are 103 unique values of *charterer*, 241 of *owner*, and 21 of *route*; leading to the need for a massive expansion of the dataset if one-hot encoding is employed. Furthermore, one-hot encoding converts the categorical variables into dummy variables with only two levels (0 and 1 referring to the presence of that category), resulting in very sparse decision trees with only two options for splitting and the tendency of growing the tree in one direction. Therefore, we opted for target encoding which possesses a clear advantage over one-hot encoding in this study. Target encoding is a Bayesian-based encoder that uses information from dependent variables to encode the categorical data. In this case, the posterior probability of the target would replace each category.

One disadvantage of target encoding is the possibility of overfitting as a result of data leakage. There are two ways to tackle this problem:

- Leave one out: This approach would exclude the target variable of the current observation while measuring the encoding value for that observation.
- K-fold: The data set is divided into k number of folds and then, k-fold cross-validation is performed to find the encoding value for each fold.

However, those mentioned methods come at the cost of losing the interpretability of the model afterward as there are different encoding values for each category. Therefore, in order to maintain the interpretability and avoid overfitting in this case, target encoding is performed on train data only and the encoding values are pasted to each matched category in the test set. In the cases that categories in the test set are not covered by the train set, the global mean value of the target variable would be the encoding value for those categories.

In general, most westward routes have lower encoded values than eastward ones. While the encoded values of the top ten charterers range from 3.6 to 4.1, those of the top ten owners are from 3.8 to 4.1[10].

[10] Appendix A2 contains details of encoded values for routes, charterers, and owners.

5.2. Fitting Models

Next, the models of GAM and XGBoost are fitted using seven determinants: *lead time, age, load factor, route, charter, owner,* and *market index*.

In order to build the hedonic price model using micro-level determinants of the freight rates for oil tanker shipping contracts in the VLCC spot market, we examine the following GAM model:

$$g(E(F_i)) = \gamma_0 + s(L_i) + s(A_i) + s(U_i) + s(I_i^R) + s(I_i^O) + s(I_i^C) + s(M_i) + \varepsilon_i \quad (10)$$

where $E(F_i)$ refers to the expected value of logarithm of the observed freight rate of the i^{th} fixture at time *t*; L_i is an abbreviation for the lead time of i^{th} fixture; A_i and U_i stand for vessel's age and utilization ratio, respectively; I_i^R, I_i^O, and I_i^C relate to the encoded values of route, shipowner and charterer, respectively; $M_i$is interpreted as the market index at month i^{th}, with $M_i \in [1, \ldots, m]$; and $\varepsilon_i$is a residual perturbation .

5.2.1. Tuning Hyperparameters

Hyperparameter tuning is thus essential to select the best parameters to make the model with better performance.

Here we thoroughly tune six parameters that usually have a big impact on the performance of XGBoost model and are outlined below:

- nrounds: controls the maximum number of trees to grow. The higher value of nrounds means higher iterations. We implement tuning for nrounds from 500 to 2000. As trees are built sequentially, by observing whether adding a new tree improves the performance of the model, we can choose the optimal value of nrounds.
- max_depth: identifies the depth of the tree or number of splits in each tree. Higher value implies a more complicated model, but also a risk of overfitting. We choose max_depth ranging from 1 to 10.
- eta: while max_depth may lead to the risk of overfitting the model, parameter eta will alleviate this issue. eta denotes the learning rate and control shrinkage of feature weights at each round. A low value of eta may cause slow computation; therefore, the model must be compensated by higher nrounds. The chosen range of eta is from 0.01 to 0.3.

- lambda: controls L2 regularization on weights and also helps to prevent overfitting. We run the tuning for lambda from 0 to 0.01.
- min_child_weight: limits the minimum number of samples in a node. The split in a node will stop and the node becomes a leaf if the sum of instance weight is smaller than min_child_weight. By that, we can reduce computable time and avoid overfitting models. The chosen range of min_child_weight is from 1 to 10.
- sub_sample: stands for the ratio of the training instance that XGBoost can randomly select to grow trees. Again, by controlling sub_sample, we can prevent overfitting and complexity of the model. The range of sub_sample for tuning is between 0.5 and 0.8.

Once making a list of parameters with their ranges, we implement blocked cross-validation and employ random search (with 10 iterations) on 4 slices of our training dataset to measure the performance of each set of the parameter and find the best parameters for the model.

The optimal parameters are listed in table 4.

Hyperparameters	Range	Optimal parameters
nrounds	500 to 2000	1452
max_depth	1 to 10	3
eta	0.01 to 0.3	0.212
lambda	0 to 0.01	0.0293
min_child_weight	1 to 10	3
sub_sample	0.5 to 0.8	0.733
MSE		0.0573

Table 4 - Overview of hyperparameters

5.2.2. Fitting Model & Evaluation Index

We then apply the optimal parameters found in the previous step to fit GAM and XGBoost models. Once two models are trained, we proceed to perform predictions on the test set. We then adopt three common indexes for continuous dependent variables to evaluate how well the two hedonic price models capture the pattern in the test set.

Root mean square error (RMSE):

While MSE (mean squared error) reflects the difference between the fitted values and the corresponding observation extracted by averaging the absolute difference, RMSE is the square root of MSE. RMSE is defined by the following formula:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \qquad (11)$$

where $\hat{y}_i$ and y_i are the fitted and the actual variables of the observation data in the test set, respectively; and n is number of observations.

Although RMSE and MSE have been widely adopted to evaluate accuracy, these two methods are sensitive to outliers. Given the fact that all differences are treated equally, large residuals affect MSE and RMSE more (Hyndman et al., 2018). Smaller RMSE implies a better fit of the model.

Mean absolute error (MAE):

MAE reflects the absolute difference between the fitted value and the actual value extracted by squaring the average difference:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \qquad (13)$$

MAE is more robust to outliers. The smaller values indicate the higher prediction accuracy and better fit of the model. The difference between MAE and RMSE is that the contribution of all individual errors to the result of MAE is linear, while RMSE ignores small values and takes more consideration in large values (Hyndman et al., 2018).

Besides these previous scale-dependent measures, we also carry out the percentage-error measure.

Mean absolute percentage error (MAPE):

MAPE calculates the mean of the absolute percentage errors and can be expressed as:

$$MAPE = \sum_{i=1}^{n} \left| \frac{(\hat{y}_i - y_i)}{y_i} \right| \times \frac{100\%}{n} \qquad (14)$$

Since MAPE is free from scale constraints of the response variable, this measure is advisable to evaluate the performance of different data. Besides, it also prevents negative and positive errors from deducting each other. However, the downside of MAPE is that if the dependent value is closer to 0 or equal 0, MAPE may become infinite or undefined so it will not be valuable in this context. Besides, it also puts heavier penalties on negative errors than positive errors (Hyndman et al., 2018).

5.2.3. XGBoost Feature Importance

Feature importance is one of the advanced applications of XGBoost. Basically, XGBoost Importance implies how beneficial explanatory variables are when contributing to building the trees. The higher importance score implies greater attribution of variables in making a decision tree. It first measures how each attribute node improves the performance of the decision tree, accountable for the number of observations of each node. The importance score is then averaged over all decision trees in the model. To facilitate the interpretation of feature importance, we sort and rank important scores of all features in descending order. XGBoost Importance provides information about the following three scores:

- Gain: represents the improvement in accuracy by adding a feature to the branches. Hence, a higher percentage means a greater contribution to the model. This is the most useful attribute to interpret the feature's importance.
- Cover: measures the relative number of times a feature appears in the trees.
- Frequency: counts the number of times a feature is employed in decision trees.

5.3. SHAP Values

Although to an extent, XGBoost Features Importance can explain the predictions from the model and thus, increasing interpretability, it can be misleading. For example, *charterer* and

owner variables have 103 and 241 levels, respectively while *load factor* has far more levels (i.e., 2014 unique values). There is a high possibility that *charterer* and *owner* are used less often in a tree while the others with the higher number of possible values might contribute more in different levels of the tree. Therefore, we use SHAP values as an alternative measurement in order to lessen this bias and compare them with the values obtained from XGBoost Feature Importance.

At a global level, SHAP values can estimate the contribution of each determinant based on the magnitude of feature attributions (which possibly results in less biased estimation compared to XGBoost Feature Importance). Moreover, with SHAP summary plot, not only the size but also the direction of the impact of specific feature value on the prediction are depicted. On the other hand, SHAP Dependence plot is a kind of partial dependence plot that shows the marginal effect of picked features on the prediction of a machine learning model (J. H. Friedman 2001). Thus, the relationship between the outcome and the feature value is revealed.

At a local level, SHAP is used to measure the contribution of each feature to the prediction of each observation, which traditional XGBoost Feature Importance is not able to do. With that information, we can easily explain the outcome, increasing interpretability or transparency of the model.

The expanded application of SHAP Dependence plot is to highlight feature interactions. To produce SHAP Dependence plot for interaction effect, first, the Shapley interaction index is computed after taking into consideration the main effect of each feature (meaning that the individual effects would be subtracted in order to obtain the pure interaction effect).

6. Results & Discussion

The chapter begins by evaluating and comparing the performance of GAM and XGBoost models before performing a variance analysis of freight rates from two hedonic price models. We continue by presenting a thorough analysis of SHAP value by means of the model with higher accuracy measures.

6.1. GAM and XGBoost

6.1.1. Accuracy Measures

The following two plots illustrate how the estimated values based on GAM and XGBoost fit the corresponding actual values of freight rates over test data. Scatterplots in figure 7 visualize the out-of-sample estimations. The 45-degree line implies the perfect scenario. Hence, observations located more closely on the line indicate better prediction. Figure 7B emphasizes the considerably positive hedonic relation between the actual and fitted values produced by the XGBoost model. Meanwhile, the fitted values measured by GAM, as illustrated in figure 7A, are scattered in a much wider range around the actual values. In general, XGBoost provides a better fit as it passes more closely while the output of GAM tends to be overestimated.

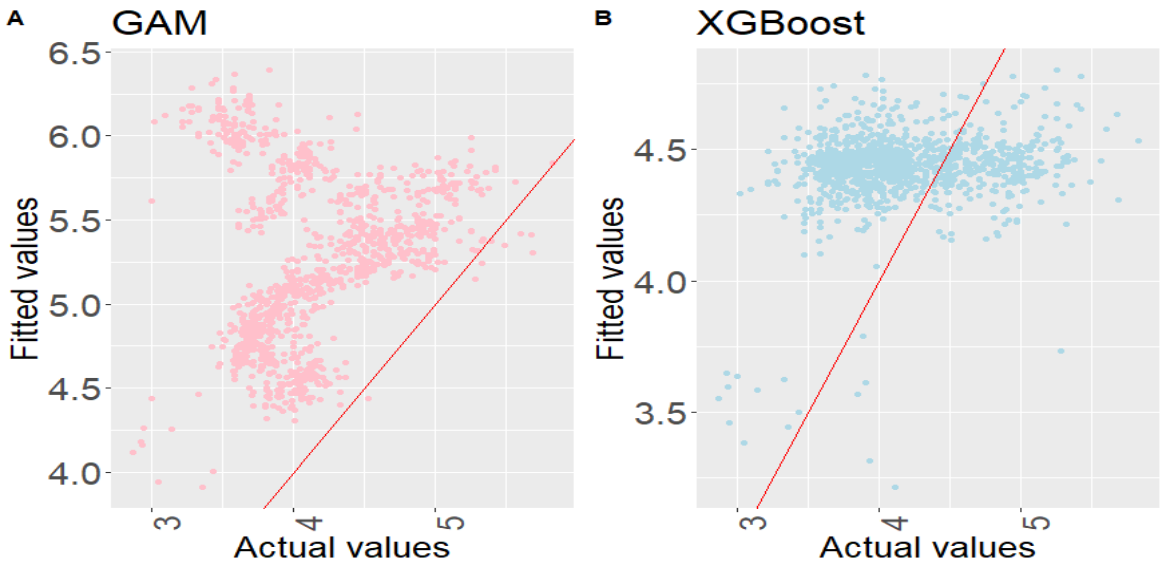


Figure 7 - Fitted vs. actual freight rate (log) by GAM and XGBoost

The evaluation results shown in table 5 provide clearer evidence for the outperformance of XGBoost over GAM. While GAM can explain 79.6% of the absolute percentage variation of the data, XGBoost proves to be superior since it can reduce MAPE by 46.1%.

Accuracy Measures	GAM	XGBoost
MSE	1.584	0.313
MAE	1.097	0.487
RMSE	1.258	0.560
MAPE	0.204	0.110

Table 5 - Model evaluation results

One possible explanation of overfitting of the models is the high variance of the test and training set. As noted in the descriptive statistics, the observations of the response variable in the training data (dataset from 2011 to 2018) are considerably consistent as opposed to the surge of the charter rates in the test data (dataset from 2019 to 2020)[11]. Due to the omission of random sampling for the sake of chronology, the risk that the model built on the training set cannot capture all patterns of the test set is unavoidable. Besides, there is possibility that microeconomic determinants cannot fully capture the volatility of charter rates since the shipping industry is also considered to be strongly driven by macroeconomic factors.

6.1.2. Results from GAM Model

Table 6 summarizes the estimated intercepts and effective degree of freedom (EDF), which reflects the degree of non-linear of the hedonic indices as well as their relative significance from model (10). The result has further strengthened our conviction that all predictor variables from model (10) are highly significant and clearly nonlinear since all values of EDF are higher than 1. Among those variables, *age* has the smallest EDF but still has an obviously non-linear relationship with freight rates. Our result is in good agreement with the findings by Dick et al. (1998), Alizadeth and Talley (2011a), and Adland et al. (2016) which addressed the quadratic relationship between vessel's ages and charter rates.

[11] The same situation is experienced with the market index variable as the training data do not contain the same observation as the test data. For this reason, we generate GAM and XGBoost models for full data (Appendix A7&A8) in order to compare with fixed effect linear regression model (Appendix A6).

Parametric coefficients		
	Estimate	Significance
Intercept	3.907	***
Smooth terms		
	Effective DF	Significance
s(Lead time)	6.546	***
s(Vessel age)	3.006	***
s(Utilization ratio)	3.764	**
s(Market index)	8.975	***
s(Charterer identity)	5.052	***
s(Owner identity)	2.717	***
s(Route)	5.813	***
REML score	-1003.7	
N	6367	

Note: Signif. codes are respectively: 0'***' 0.001'**' 0.01'*' 0.05'.' 0.1" 1.

Table 6 - Smooth terms from GAM model

In an attempt to better visualize the relationship between each dependent variable and the variance of charter rates, smooths, and partials from GAM model are presented in figure 8. Both *lead time* and *market inde*x have strong non-linear relationships with respect to charter rates. Generally speaking, rates fractured consistently when lead time is longer, however, the overall upward trend of rates can be still observed. Holding the other variables fixed, after the vessel hits 15 years old, the older the vessel is, the lower the rate is, although the changing of the price is quite negligible. When it comes to the relationship with vessel utilization ratio, rates go uphill slightly before load factor reaches 0.9, which is the point when most fixtures take place and fall down gradually after that. It is worth mentioning that the general market level has lower degrees of uncertainty than other features, illustrated by lower confidence bands in its curves; and confidence bands of four figures tend to be narrower around the points with higher data density.

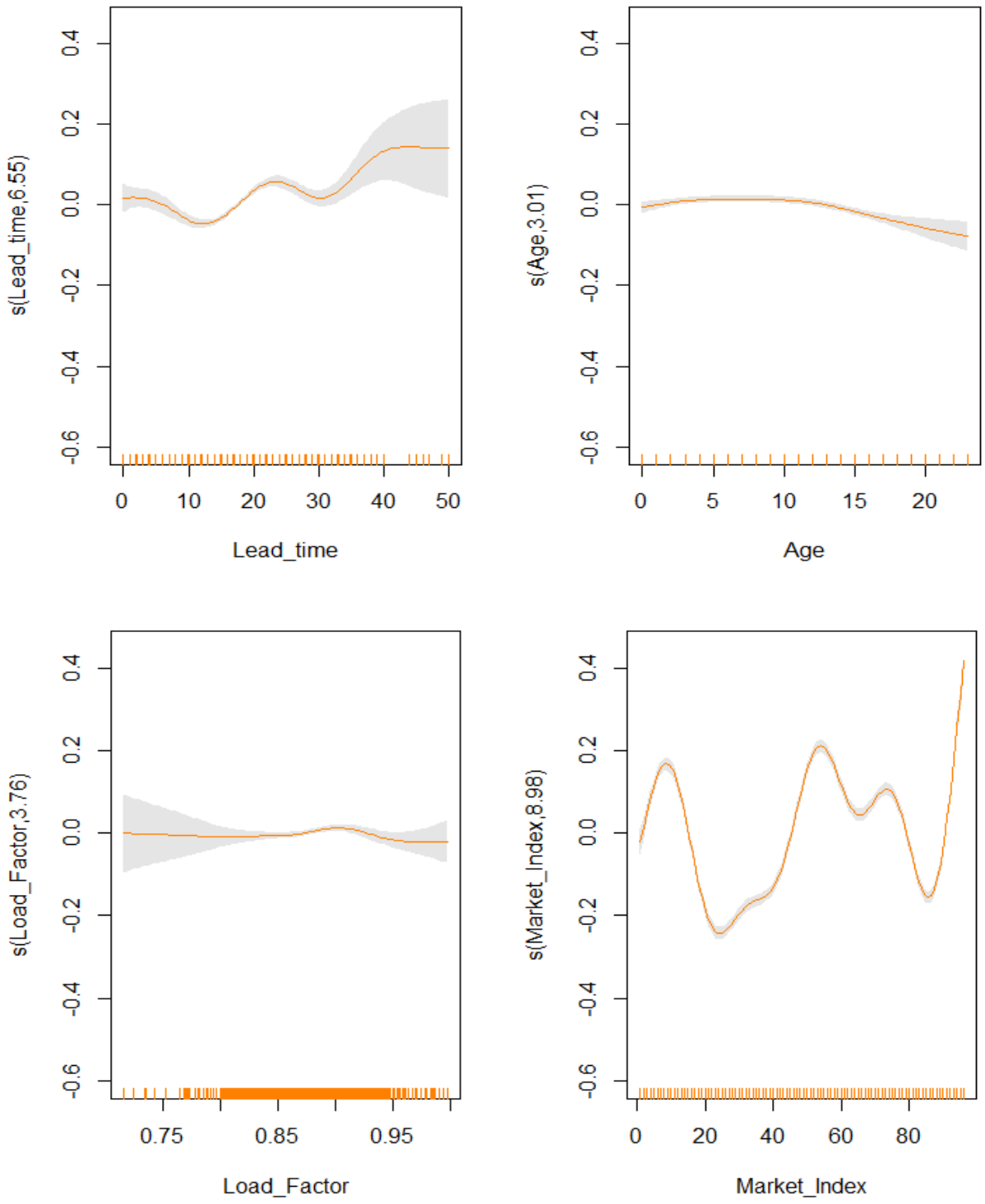


Figure 8 - Smooth of GAM model

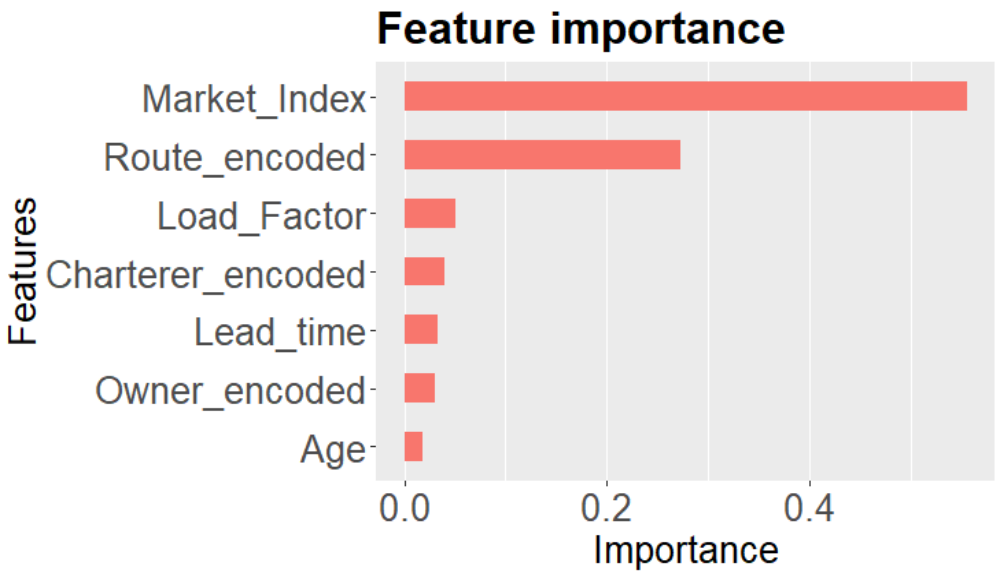
6.1.3. Results from XGBoost model

In this part, we analyze deeper into how the hedonic model is built, or in other words, how useful each factor is to make key decisions to build trees in XGBoost model.

As depicted in figure 9, market index has the largest impact since it contributes to 55.63% accuracy of XGBoost model, twice the contribution of the next feature. These results provide confirmatory evidence that the general market dominates in price formation, and additional vessel and contract variables matter much less.

Importantly, we find that taking *route* into account has a substantial impact as the explanatory power of the model increases to 27.22%. This gain score of trading routes obviously far outweighs other features. The usefulness of *routes* to explain the variation in freight rates is of no surprise since the distance between origin and destination impacts strongly on fuel consumption and inventory cost, which in turn have implications for rate levels.

The next five features have relatively low contributions for boosted trees but at the same time, there is not much difference among those indicators. Remarkably, the analysis did not show any significant effect of vessel age in price formation (by adding age to the branches, the accuracy of the model increases by only 1.66%). This finding is in line with Tamakis and Thanopoulou (2000) which rejected the rate premium paid for younger than older vessels.



Note: XGBoost feature importance scores are based on gain scores in which gain scores of all features sum up to 1 (or 100%)[12].

Figure 9 - Feature importance based on XGBoost framework

The next practical approach is to visualize the functional relationship between charter rates and each independent variable using partial dependence plots as such we can effectively compare the outcome of XGBoost with that of GAM model. In general, rates vary significantly with respect to increasing values of VLCC tanker index and encoded values of *routes* as opposed to the very partial dependence on other features, especially vessel age does not show a statistically significant impact to price volatility. Hence, the overall trend of price variation by XGBoost model closely follows the result from GAM model.

Empirical results from figure 10, 11 and 12 provide further evidence for the notion that charter rates of the VLCC spot market are highly sensitive to the changes in the tanker market conditions. As anticipated, we also observe the cyclical behavior of the tanker market with the cycle duration is approximately 5 years (or within 60 months as illustrated by figure 10).

More interestingly, although prices are monotonic increasing with increasing encoded values of routes, there are two clusters of price variance corresponding to encoded values of less than 3.75 (represents for trading routes: Persian Gulf - West, Persian Gulf - USG, Persian Gulf -

[12] Other XGBoost importance scores are provided in Appendix A4.

UKC and Persian Gulf - Canada) and more than 3.75 (the remaining cargo flows). In other words, the four backhaul routes starting from Persian Gulf to the west are significantly cheaper than other routes.

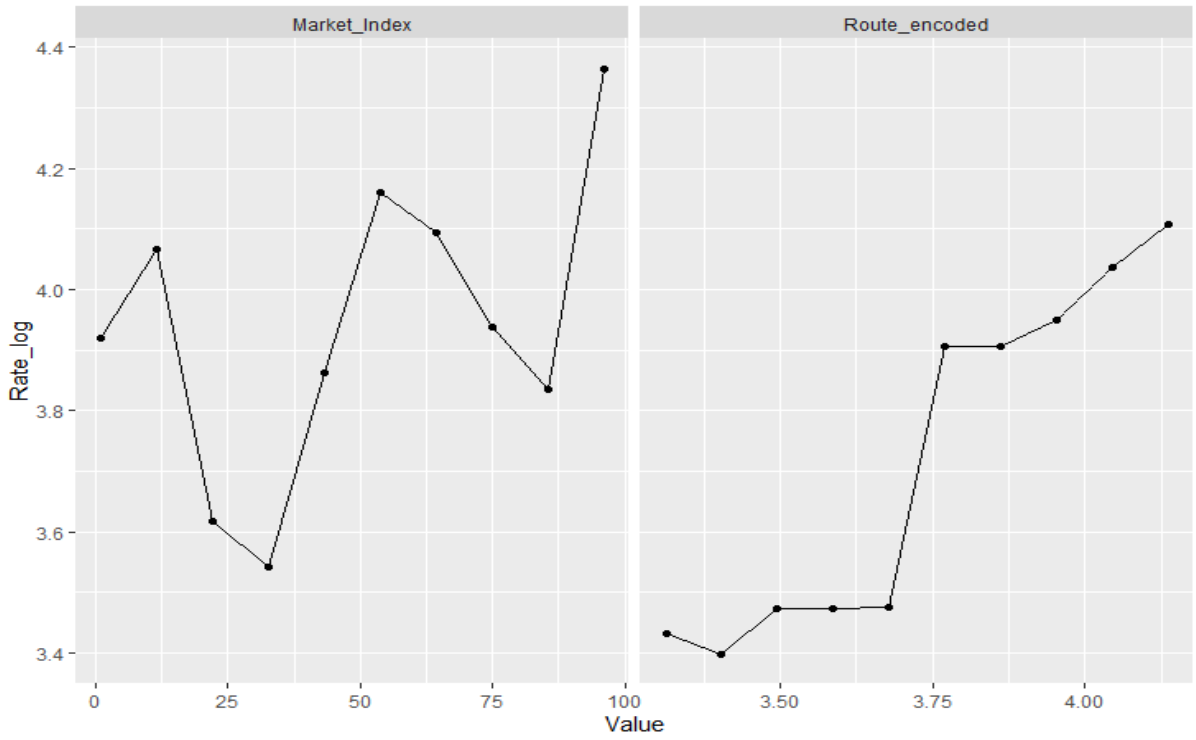


Figure 10 - Partial dependence plots of market index and route (encoded) from XGBoost model[13]

Figure 11 depicts the correlation between charterer and price variation. More specifically, charter rates are quite consistent among charterers with encoded values less than 3.4. Interestingly, this is the list of charterers who do not belong to the top 10 charterers indicated in table 2 (Top 10 charterers is a list of charterers that fixed a majority number of contracts in the observed decade). Following this point is the combination of active and less active charterers, and charter rates also fluctuate more widely. This observation suggests that the charterers might have considerable bargaining power on the negotiated freight rate. On the other hand, there is no significant evidence for the substantial influence of different ship owners to

[13] See encoded values of route in Appendix A2

volatility of price. The higher contribution of charterers over owners in our study is consistent with results from Adland et al. (2006) and Regli (2019)[14].

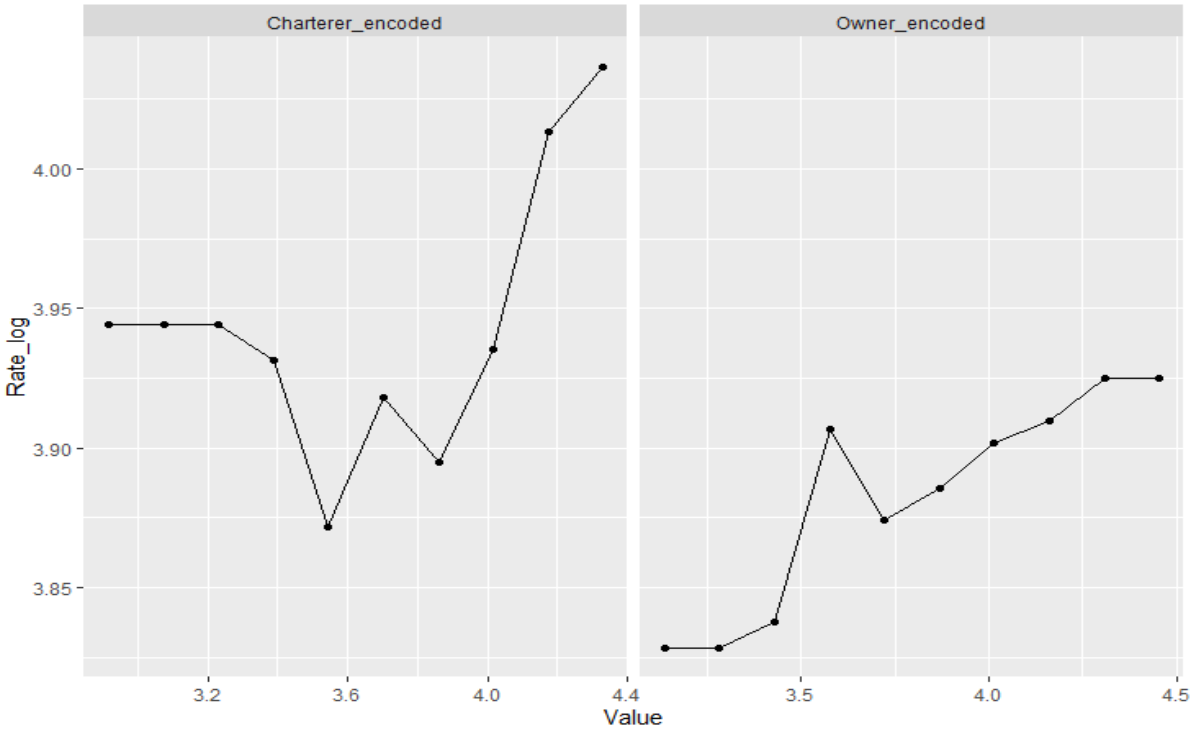


Figure 11 - Partial dependence plots of charterer and owner (encoded) from XGBoost model[15]

Besides, similar to the specific variables of charterers and owners, *lead time* also has a positive relationship with tanker freight rates although its effect is not as sensitive. In contrast, freight rates tend to decrease with *load factor* and *age* although vessel age doesn't have much effect on the price variation. More specifically, for the vessels that have utilization ratios below 0.9, fixture rates have the tendency to decrease sharply with the increasing value of load factor. Above this threshold, charter rates tend to go in the opposite direction. Meanwhile, the downward trend of price is more profound after vessels reach their 15-year-old anniversary. Combined with the result from GAM model, our findings appear to be well substantiated with insights drawn from fixed effect linear models proposed by Adland et al. (2006) and Alizadeh and Talley (2011a).

[14] Regli (2019) evaluated the bargaining power on the VLCC time charter market on the route from Persian Gulf to Far East and suggested that shipowners' bargaining power coefficients is 24%, which is supportive of the view that charterers have more bargaining power than shipowners.

[15] See encoded values of charterer and owner in Appendix A2.

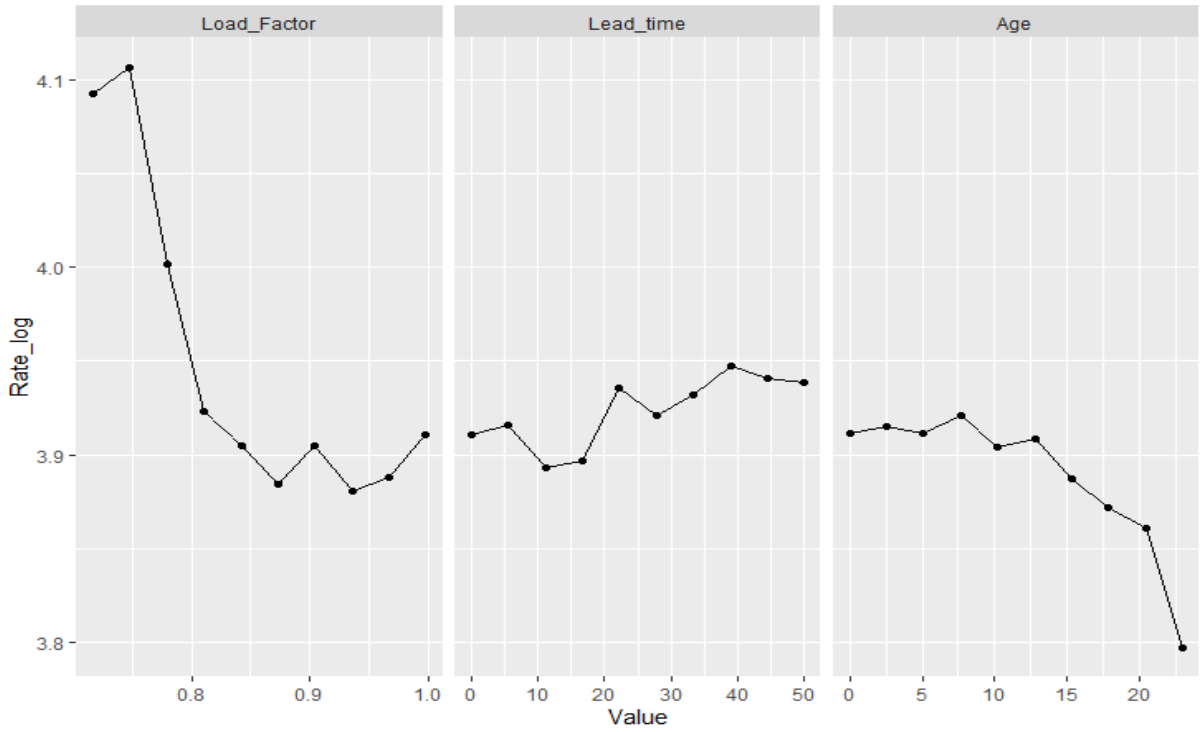


Figure 12 - Partial dependence plots of load factor, lead time and vessel age from XGBoost model

6.2. SHAP Value

Using SHAP values to obtain the contribution of each determinant, we can enhance the transparency of XGBoost model. As mentioned, SHAP values provide explanations for global and local context and interaction effects.

6.2.1. Global Explanation

Subsequent is the summary plot of SHAP values for XGBoost model (figure 13) with each dot representing a data point. The ranking of the contribution of each variable to the predictions is almost similar to that obtained from XGBoost Features Importance presented in figure 9, except the change in the order of charterer and load factor, owner, and lead time variables. The subsequent dependence plots of SHAP values for each feature depict in detail the relationship of corresponding specifications and the expected freight rate.

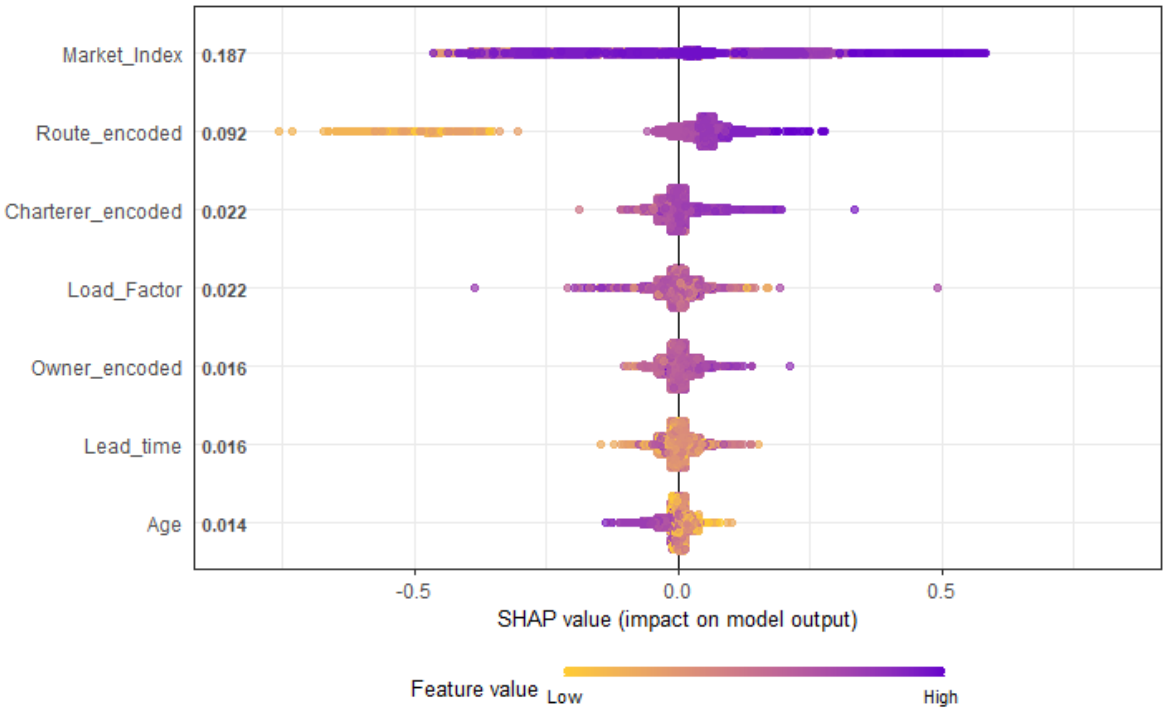


Figure 13 - Summary plot of SHAP values of each variable on the predictions

Market Index

As depicted in figure 14, the contribution of the market index variable is following a cycle pattern of approximately 60 months, or almost 5 years, which is consistent with the shipping cycle of the market. In other words, in the collapse or trough stage of a cycle, the relative impact of the market level to freight rate is negative and great in magnitude while it positively and substantially affects freight rates when the market cycle is at its peak stage. This result is aligned with the finding of Adland (2017b) which estimates a hedonic pricing regression to generate a market index from vessel characteristics and contractual terms and concludes that around 70–80% of variation in day rates is explained by the time fixed effects.

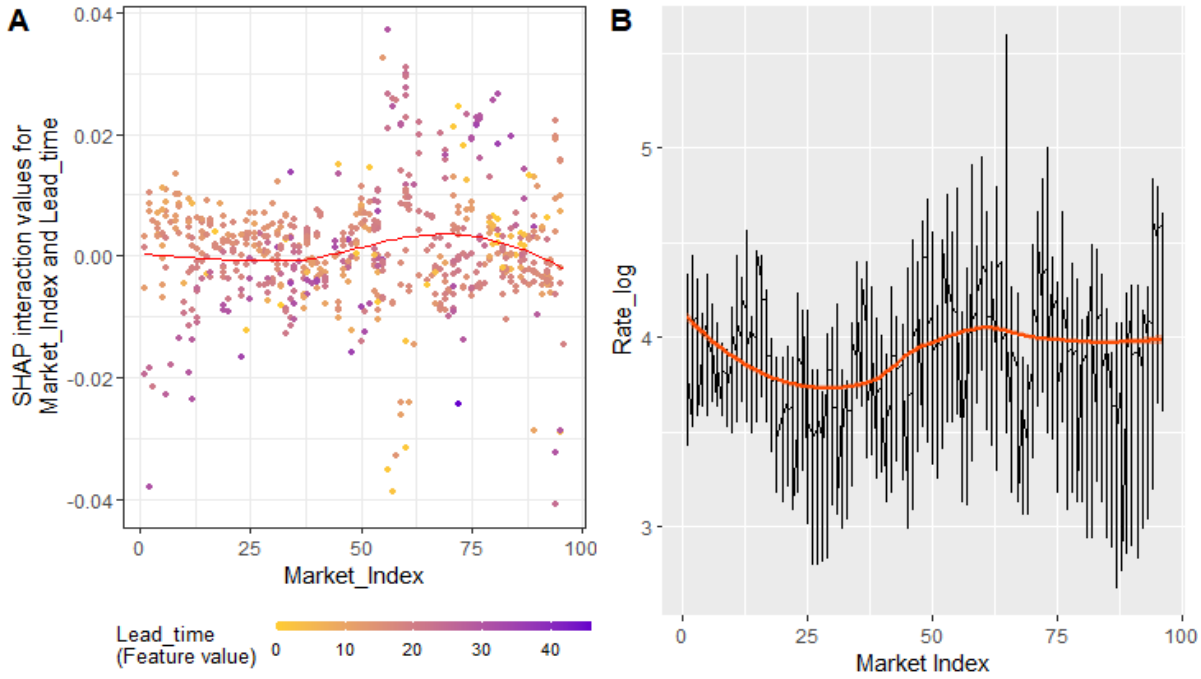


Figure 14 - SHAP value for market index and freight rate over time (at monthly level)

Route

As depicted in figure 15, the freight rate is sensitive to the route that the vessel takes. The impact of route on the expected freight rate can be classified into two groups:

- Negative impact: including four routes that have encoded values lower than 3.6 (i.e., Persian Gulf-UKC, Persian Gulf-Canada, Persian Gulf-USG, Persian Gulf-West).
- Positive impact: the rest of the observed routes.

The similarity shared among the four routes consisted in the first group is their westwards direction, in line with our initial observation that westbound routes generally have lower rates than their eastbound counterparts and the empirical results of Alizadeh and Talley (2011a). It can be explained as a strategic behavior in the VLCC charterer market that vessel owners are willing to lower their rates on those routes in order to optimize their overall revenue on the way back to the East by getting fixed in West Africa, the Caribbean or the North Sea.

The results of SHAP value for route are consistent with the previous result in XGBoost Partial Dependence Plot (figure 10).

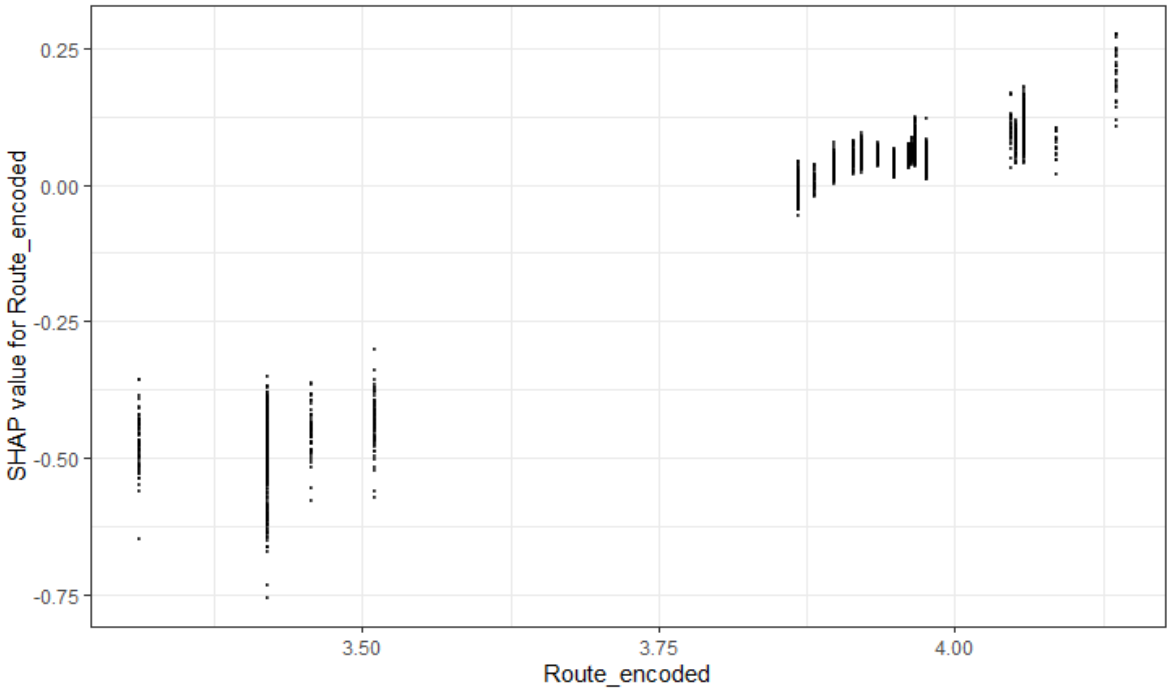


Figure 15 - SHAP value for route (encoded) for each route

No.	Route	Route_encoded	SHAP Values	Fixtures
1	WEST AFRICA_UKC	4.13674	0.20604	26
2	OTHER	4.04713	0.10605	32
3	WEST AFRICA_USG	3.96713	0.10230	45
4	PERSIAN GULF_INDIA	4.05811	0.10093	825
5	WEST AFRICA_EAST	4.05157	0.07971	123
6	WEST AFRICA_INDIA	4.08648	0.07259	16
7	WEST AFRICA_CHINA	3.96659	0.06735	692
8	WEST AFRICA_TAIWAN	3.96414	0.06239	64
9	PERSIAN GULF_SINGAPORE	3.92157	0.05410	293
10	PERSIAN GULF_JAPAN	3.96121	0.05322	320
11	PERSIAN GULF_MALAYSIA	3.9141	0.05289	72
12	PERSIAN GULF_THAILAND	3.93523	0.05235	382
13	PERSIAN GULF_CHINA	3.97658	0.04369	1155
14	PERSIAN GULF_EAST	3.94912	0.03969	278
15	PERSIAN GULF_TAIWAN	3.89815	0.03319	319
16	PERSIAN GULF_SOUTH AFRICA	3.88166	0.00570	57
17	PERSIAN GULF_SOUTH KOREA	3.86814	(0.00637)	1079
18	PERSIAN GULF_UKC	3.51067	(0.43650)	82
19	PERSIAN GULF_CANADA	3.45713	(0.45231)	59
20	PERSIAN GULF_WEST	3.31211	(0.47137)	64
21	PERSIAN GULF_USG	3.41967	(0.49443)	384

Table 7 - SHAP values of routes

Charterer and Owner

Adland et al. (2016) concludes that the characteristics of charterers and owners are significant microeconomic determinants of the freight rate level. The results from SHAP values further suggest that the contribution of charterers ranks more significantly than that of owners as the charterer variable has higher SHAP values.

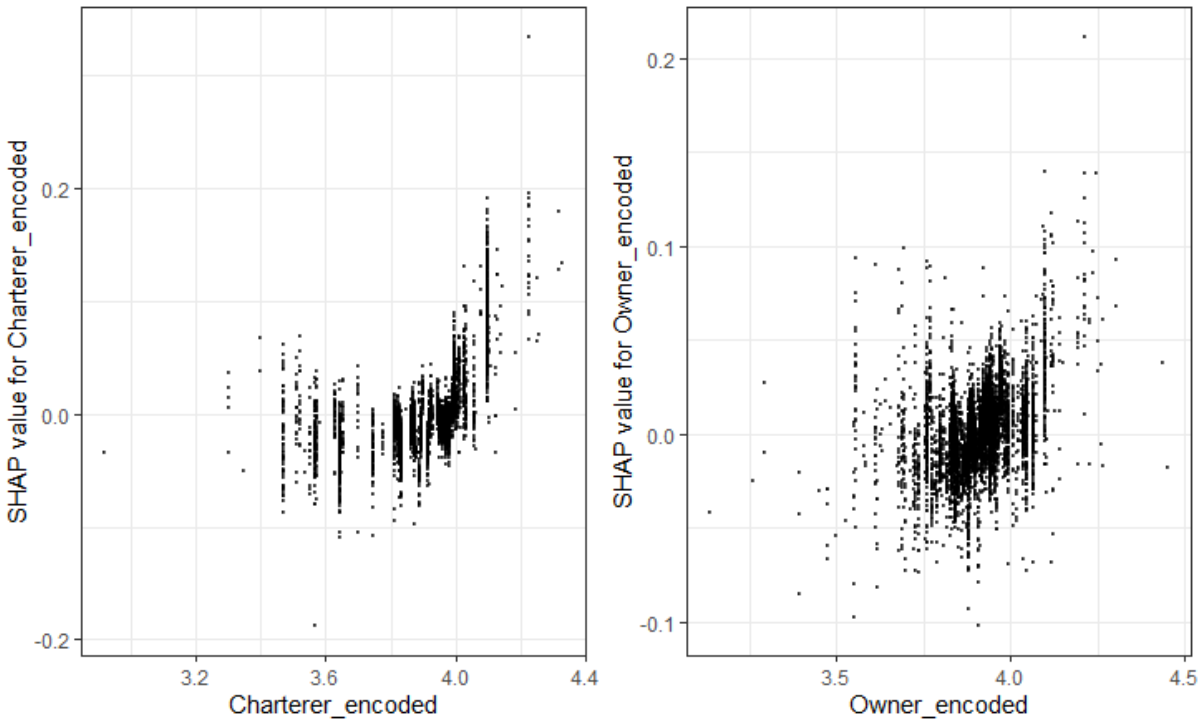


Figure 16 - SHAP values for charterer and owner

Obtained SHAP values are varied among different charterers and owners. Fixture rates are agreed upon by both charterers and owners. While charterer heterogeneity is associated with their bargaining power, owner heterogeneity is highly related to the specifications of their vessels (Adland et al., 2017b). All top ten charterers, except IOC and Reliance, have negative average SHAP values on freight rate, indicating that those charterers have leverage on the negotiating freight rate. Moreover, the magnitude of their contributions is greater than the average SHAP values of all charterers (-0.000804), implying that 8 of the top 10 charterers possess substantial bargaining power to decrease the fixture rates in comparison with their counterparts.

No.	Charterer	SHAP Values	Fixtures
1	UNIPEC	(0.00430)	1062
2	IOC	0.08520	429
3	CHEVTEX	(0.01080)	427
4	S.OIL	(0.03020)	382
5	PTT	(0.00520)	329
6	DAY HARVEST	(0.00990)	310
7	SHELL	(0.01820)	286
8	EXXONMOBIL	(0.03500)	266
9	RELIANCE	0.02990	247
10	CPC	(0.01540)	218

Table 8 - SHAP values of top ten charterers

On the other hand, the top ten owners, except for Maran Tankers Mngt., Euronav NV, Shpg Corp of India, have a positive contribution to fixture rates. The average SHAP values of owners is 0.00153.

No.	Owner	SHAP Values	Fixtures
1	Maran Tankers Mngt.	(0.00391)	193
2	Euronav NV	(0.00259)	191
3	New Shipping	0.00654	181
4	DHT Management	0.01647	167
5	Dynacom Tankers Mngt	0.03040	165
6	Aeolos Management	0.00316	162
7	Maran Tankers Mgmt	0.02353	146
8	Ocean Tankers	0.00065	135
9	Mitsui O.S.K. Lines	0.01510	127
10	Shpg Corp of India	(0.01246)	122

Table 9 - SHAP values of top ten owners

Lead time

Regarding SHAP value for lead time, there is no clear pattern of how lead time contributes to the predicted freight rate. It can be explained by the simultaneous interaction between lead time and the freight rate. To be specific, Alizadeh and Talley (2011b) concludes that lead time and dry bulk freight rates are interrelated and determined simultaneously; and the estimated results for the tanker market in Alizadeh and Talley (2011a) suggest that ships are fixed earlier during times of high freight rates and lower volatility. Prochazka et al. (2019) derives a similar conclusion that oil buyers secure tonnage earlier during strong tanker markets. Moreover, the

paper suggests that the geography of trade creates natural decision points that dominate the spatial distribution of fixtures, which in turn, affects lead time.

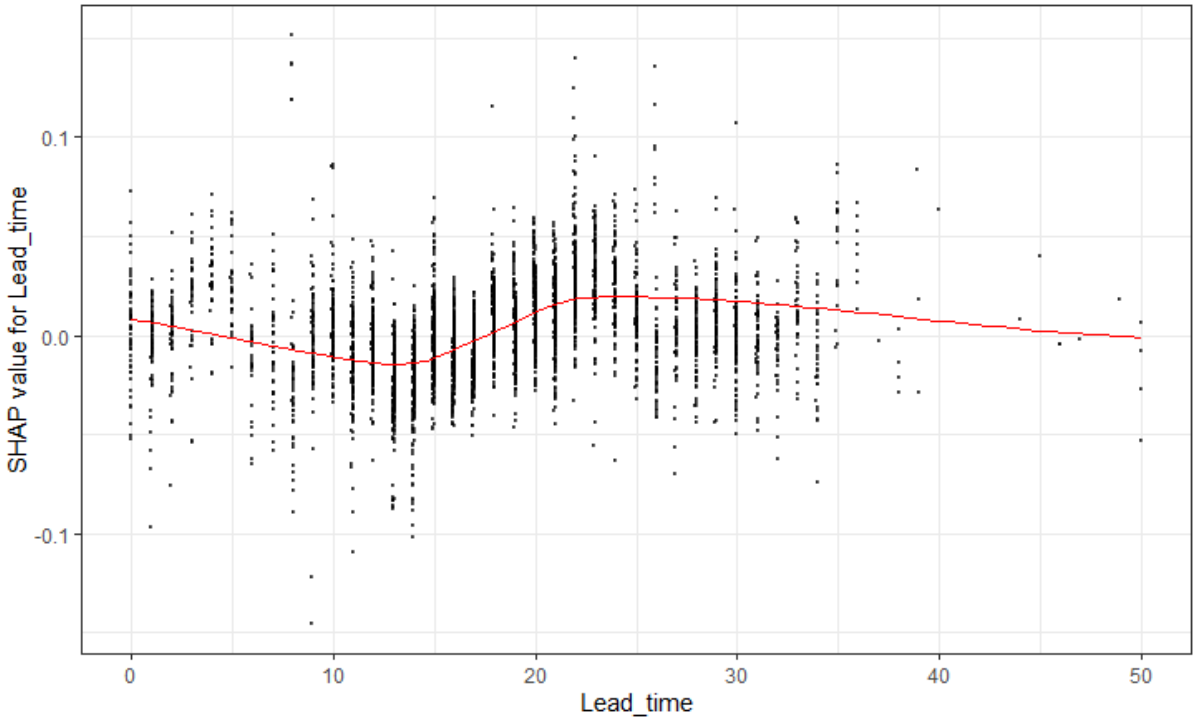


Figure 17 - SHAP value for lead time

Load Factor

Based on the theory of economies-of-scale, it can be argued that when the load factor increases, the marginal cost of transporting one extra unit has a tendency to decrease, lowering the expected freight rate. Adland et al. (2016) and Olsen et al. (2017) confirm this expectation as they found the inverse relationship between utilization ratio and freight rate. On the other hand, it is also reasonable to argue that a high utilization ratio implies a shortage in supply, which in turn, increases the expected freight rate. Figure 16 illustrates the inconsistent influence pattern of load factor on fixture rates. For the vessels that have load factors below approximately 0.9, utilization ratio seems to negatively impact the fixture rates, although the magnitude of the influence is not significant. Above this threshold, there are positive SHAP values for load factor, supporting the latter reasoning.

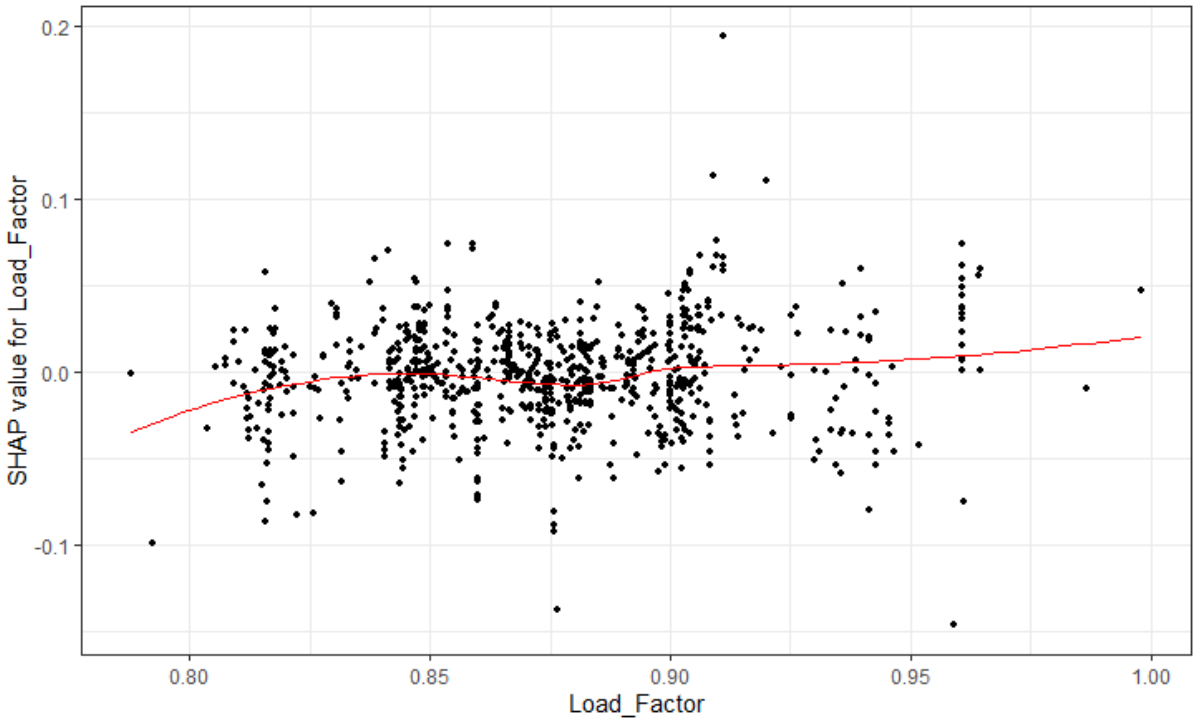


Figure 18 - SHAP value for load factor

Age

Based on figure 19, age of a vessel tends to have a positive but insignificant impact on the estimations of freight rate until the vessel reaches an age of approximately 14-15 years. Exceeding this threshold, the vessel age negatively contributes to its fixture rate and the magnitude of this impact escalates as the age increases. Our finding is consistent with the results from Alizadeh and Talley (2011a) and Adland et al. (2016) that age tends to have a positive impact with small magnitude to the freight rate while age squared, which represents the quadratic relationship, is negatively correlated with the freight rate with the minimum age of 15 years. On the other hand, Tamvakis and Thanopoulou (2000), based their investigation on the data from 1989 to 1996 found no significant impact of age on freight rate. The difference in results of these mentioned papers may associate with the observed periods, which cover different stages of the shipping cycle.

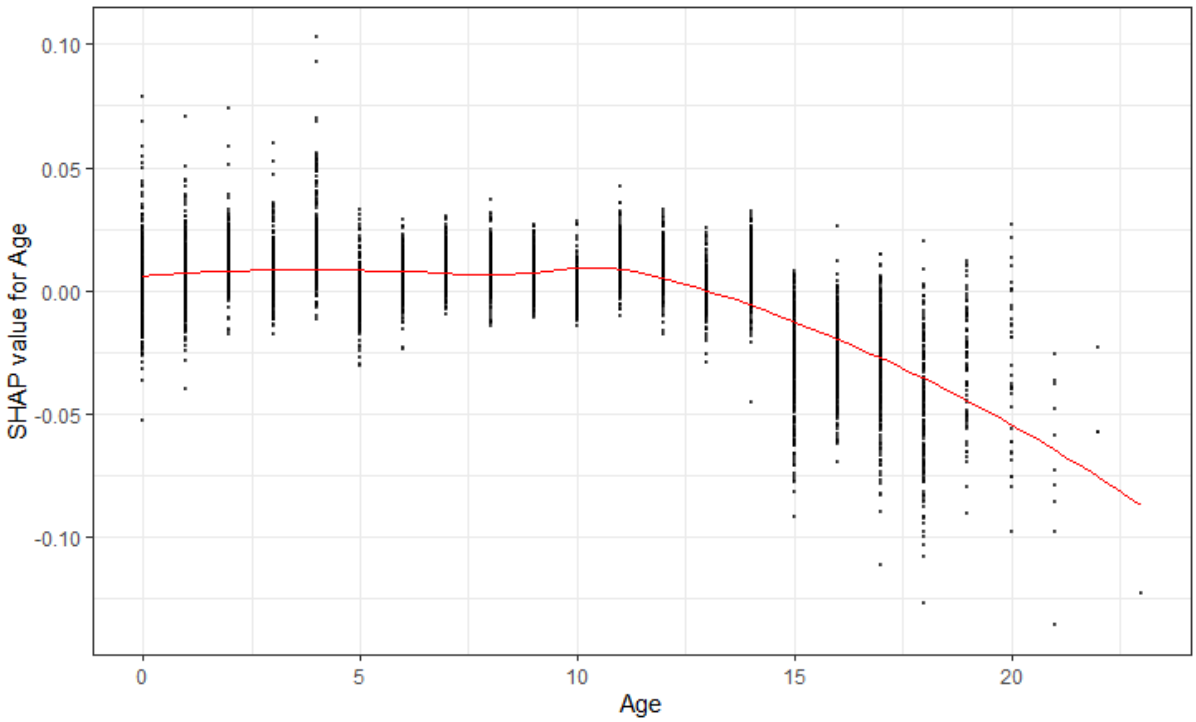


Figure 19 - SHAP value for age

6.2.2. Interaction Effect Explanation

The advantage of SHAP value is the ability to efficiently measure every possibility of interaction between a pair of variables. Conducting experiments on all possible pairs of variables, we find significant interactions between route and other variables. Furthermore, the interaction between charterer and owner, which is mentioned in Adland et al. (2016), is also investigated and compared with the finding of the mentioned literature.

Route x Charterer

As can be seen in figure 18, the SHAP interaction values for Charterer x Route and Owner x Route are varied among charterer and owners. The differences are more significant in the case of charterers. In particular, the combination of charterers with encoded values below 3.9 (e.g. CPC, Chevtex, S.Oil, Shell, ExxonMobil) and routes with encoded values lower than 3.6 and demonstrated by dots in shades of yellow (i.e. four westbound routes: Persian Gulf-UKC, Persian Gulf-Canada, Persian Gulf-USG, and Persian Gulf-West) generates a negative contribution to fixture rates. On the other hand, interactions between the routes with higher encoded values (i.e., demonstrated by dots in purple shades), and charterers with encoded

values higher than 4.0 (e.g., IOC) are more likely to impact freight rates in a positive direction. One possible explanation is that the routes are associated with a charterer's supply chain, leading to the domination of some charterers on specific routes. Therefore, influential charterers on westbound routes tend to have a negative contribution to the freight rate. For instance, based on our train data (2011-2018), ExxonMobil can be regarded as a dominant charterer on Persian Gulf-USG as it accounted for 23.4% of transactions on this route. The average SHAP value of ExxonMobil on this route is -0.0416, which is higher in magnitude compared to the average SHAP values for this charterer on all routes (-0.035). On the other hand, IOC is the most active charterer on Persian Gulf-India route as it signed a majority proportion of contracts on this route (51.6%). The average SHAP values for this charterer on the route are 0.0855, which is higher than its average SHAP values on all routes (0.0852) and the average SHAP values of all charterers in Persian Gulf-India route (0.062). In conclusion, while the dominant charterers on westbound routes tend to have the bargaining power to lower the fixture rates on those routes, the influential charterers on eastbound routes are willing to pay more for the transactions of eastward voyages.

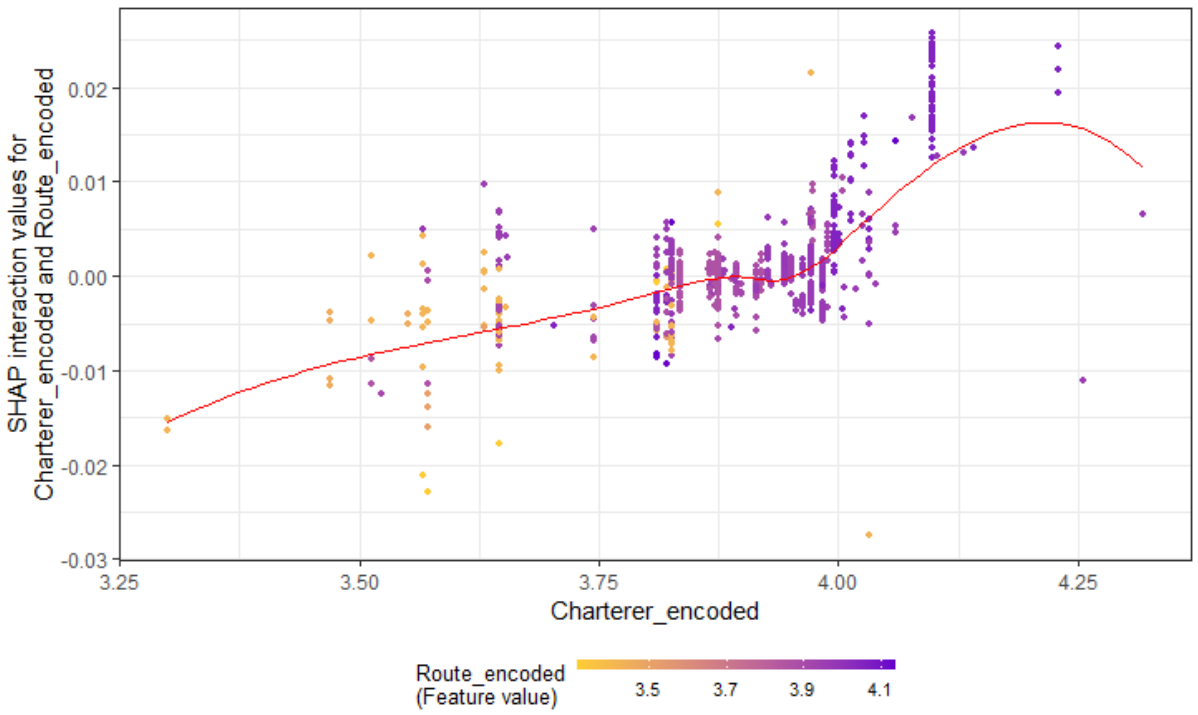


Figure 20 - SHAP interaction value for charterer and route

Age x Route

The SHAP values for the interaction between age and route implies the choice of vessels based on their ages on different routes. As can be seen in figure 21, for the westwards routes that have encoded values lower than 3.6 (i.e., Persian Gulf-UKC, Persian Gulf-Canada, Persian Gulf-USG, and Persian Gulf-West), vessels with age below 10 years old (i.e., demonstrated by dots in yellow shades) are often chosen. This is opposed to no obvious pattern of choice in the rest of the routes. On those four westbound routes, the SHAP interaction values for route and age are more scattered than other routes (i.e., higher values for newer vessels and lower scores for the older). In some extreme cases, when vessels older than 10 years old are used, the SHAP values are negative and greater in magnitude in comparison with the same circumstances on other routes. Those observations imply that charterers consider vessel age when fixing a contract on westbound routes. The positive average SHAP interaction values also indicate the willingness to pay more for newer vessels on mentioned westbound routes.

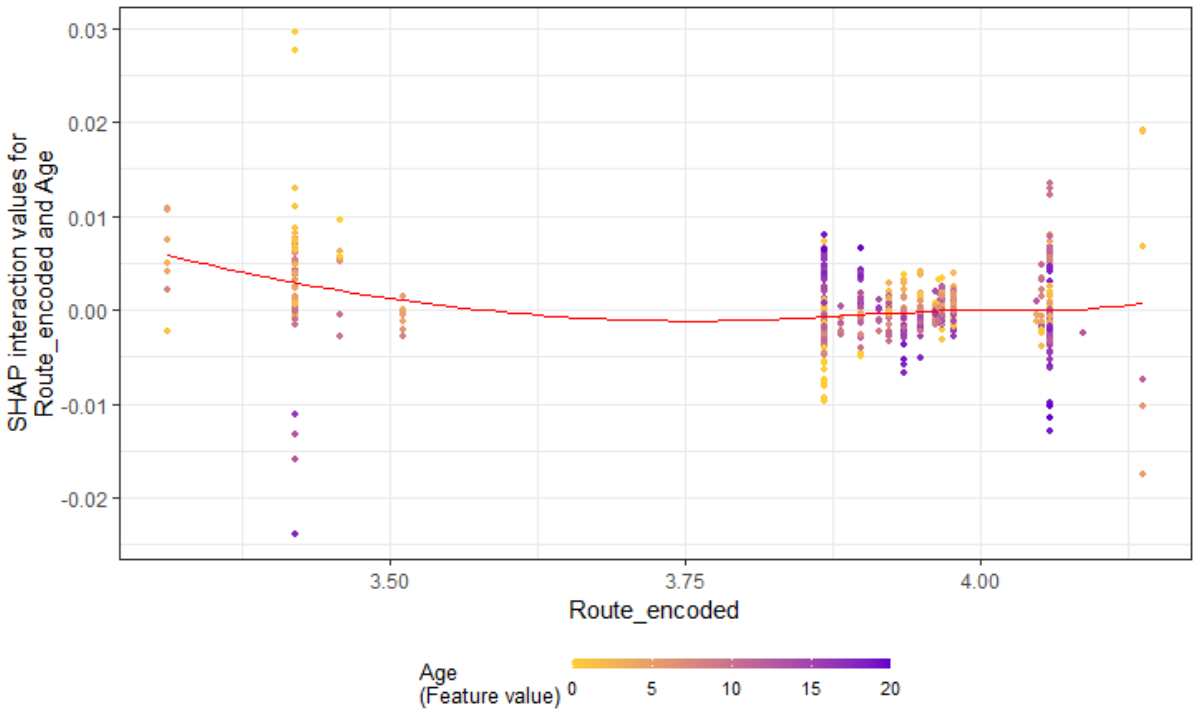


Figure 21 - SHAP interaction value for route and age

Owner x Charterer

Figure 22 depicts the SHAP interaction values for owner and charterer. In general, although there is an interaction between owner and charterer, the interaction impact on fixture rate is

insignificant. This is in line with the empirical results of Adland et al. (2016) which highlight the marginal impact of the match effect of charterers and owners on the freight rates.

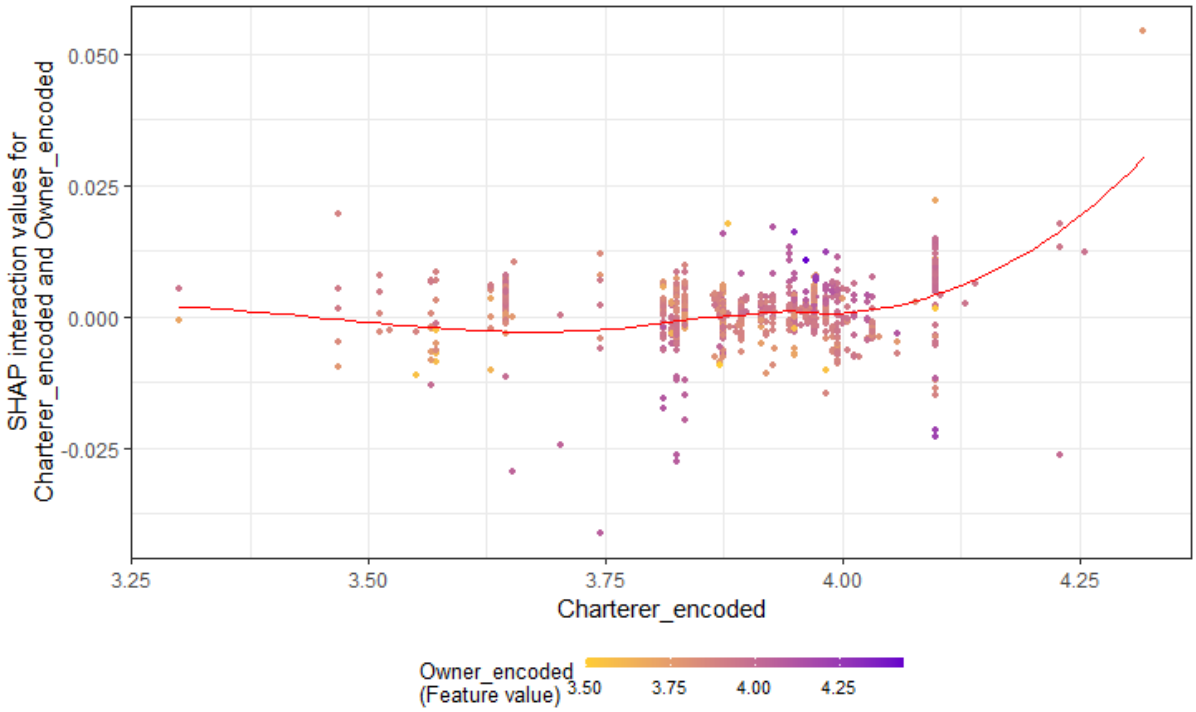


Figure 22 - SHAP interaction value for owner and charterer

Lead Time x Market Index

As mentioned before, there have been two notable pieces of literature that investigate the relationship between lead time, market index and freight rate level. Alizadeh and Talley (2011a) concludes that ships are fixed earlier during times of high freight rates and lower volatility and a similar conclusion is reached by Prochazka et al. (2019) that oil buyers secure tonnage earlier during strong tanker markets. Our results are also consistent with those previous findings, as can be seen in figure 23. Along with the rise freight rate in 2015, the impact of interactions between lead time and market index on fixture rates is positive and higher than the previous period.

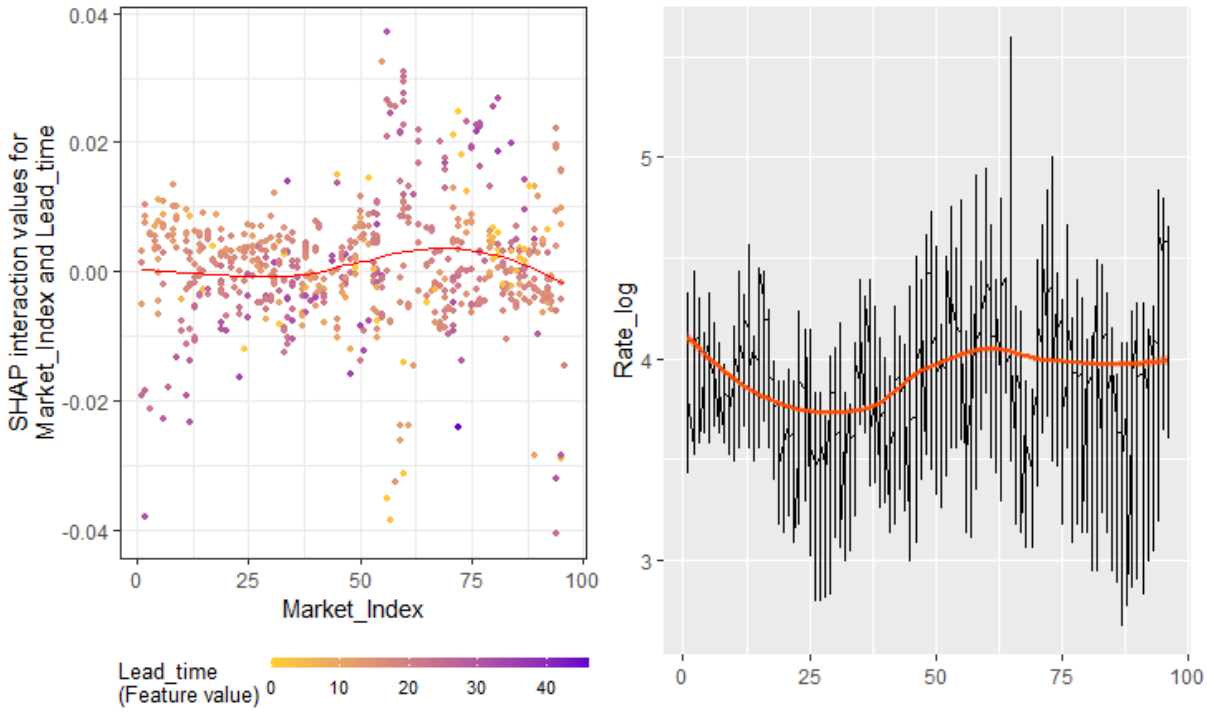


Figure 23 - SHAP value for market index and freight rate over time (at monthly level)

6.2.3. Local Explanation

On the local level, SHAP values are able to provide a detailed measurement of each variable's contribution to an individual estimation. The mean estimation for freight rate is 3.9074 (equivalent to a fixture rate of WS49.77).

As route is the second most influential determinant to the estimations, we used route as the criterion to select four examples (among 1,118 observations in the test set) to examine the impact of each feature value on four chosen observations. Intending to provide a better overview of how estimated rates are derived from SHAP values, we pick the 4 transactions that were fixed on two routes:

- Persian Gulf-India: representative for eastbound routes with SHAP values that are positive and significant in magnitude.
- Persian Gulf-USG: representative for westbound routes with SHAP value that are negative and significant in magnitude.

No.	Charterer		Owner		Route		Lead Time		Age		Load Factor		Market Index		Rate	
	Name	SHAP	Name	SHAP	Name	SHAP	Value	SHAP	Value	SHAP	Value	SHAP	Value	SHAP	Observed	Estimated
1	IOC	0.044	Shpg Corp of India	(0.030)	PG_INDIA	0.098	19	0.001	10	0.014	0.83	(0.007)	99	0.433	71	86.6
2	RELIANCE	0.076	Dynacom Tankers Mgmt	0.064	PG_INDIA	0.101	18	(0.020)	17	(0.030)	0.88	(0.007)	104	0.479	85	96.6
3	EXXONMOBIL	(0.076)	Euronav NV	(0.017)	PG_USG	(0.610)	13	(0.007)	3	0.010	0.94	(0.059)	107	0.312	50	31.8
4	SHELL	(0.004)	Frontline	(0.004)	PG_USG	(0.617)	16	0.010	10	0.020	0.94	(0.006)	107	0.320	46	37.6

Table 10 - Examples of contribution of each variable to individual predictions (4 out of 1,118 total observations)

Table 10 contains 4 observations from the data with the true values and SHAP values for each feature, the observed and estimated rate measured by XGBoost. IOC and Reliance are both major charterers on the route from Persian Gulf to India with their proportion of transactions are 51.6% and 24.7%, respectively. As explained in the section of interaction between charterer and route, the influential charterers of eastwards routes are willing to pay higher fixture rates on those routes. Correspondingly, SHAP values of the two charterers are positive, implying that the charter identities of IOC and Reliance in this specific case are associated with an increase of 0.044 and 0.076, respectively, to the mean estimated logarithm of rate. The SHAP value for the owner Shpg Corp of India generates a decrease of mean estimated by 0.03 and the SHAP value of Dynacom Tankers Mgmt means an increase of 0.064. While the first transaction has a positive SHAP value for age as the chartered vessel is 10 years old, the value for the second observation is negative as a result of a vessel age of 17, which is greater than the mentioned threshold of 15. Similar logic can be applied to explain the contribution of lead time, load factor, and market index. By adding the SHAP values of all features to the mean estimated, the logarithms of rates for the first two examples are 4.46 and 4.57, equivalent to rates of 86.6 and 96.6 WS, respectively. The estimated rates are 21.95 % and 13.6% higher than the actual values.

On the other hand, both ExxonMobil and Shell obtain negative SHAP values for their identities. As mentioned, ExxonMobil is the dominant charterer on Persian Gulf-USG route. Its SHAP values is not only negative but significant in magnitude, implying its bargaining power as an influential charterer on this westbound route. Load factor of 0.94 generates negative SHAP values as discussed. The estimated logs of rate for third and fourth observation are 3.46 and 3.62, equivalent to rates of WS31.8 and WS37.6. The estimated rates are 36.4 % and 18.3% lower than the actual values.

The same technique can be employed to interpret each individual estimation. Thus, SHAP values is an efficient and reliable tool to assess the fixture rates in individual contracts.

7. Conclusion

Our study uses advanced models (i.e., GAM and XGBoost) to conduct a hedonic price model using microeconomic determinants for the VLCC market and SHAP values to explain the influence of explanatory variables on the estimation of individual fixture rate. In conclusion, XGBoost performs better than GAM as XGBoost reduces MAPE by 46.1%. XGBoost Features Importance and SHAP values deliver almost the same results of contribution ranking of each variable on the outcomes in general with a slight change in the order of load factor, owner encoded value, and lead time variables. Market condition and cyclicality have the greatest impact on the estimations, following by route and charterer. The heterogeneity of route influencing model output reflects strategic behavior in the VLCC charterer market. Our estimated results also suggest that charterers have considerable leverage on the freight rate in this market, compared to that of owners. Most of our empirical results from SHAP values are in line with the findings of previous literature. Furthermore, SHAP interaction values suggest that influential charterers on westbound routes have the bargaining power to reduce the fixture rates while their counterparts on eastbound routes are willing to pay a higher price than mean estimation. The interaction between owner and charterer, lead time, and market level follows the results from previous notable papers. Finally, SHAP value is an efficient tool to investigate the contribution of each determinant on individual estimations, shedding light on the black box model of XGBoost.

In the course of this study, we acknowledge the following constraints and biases. First of all, the study concentrates on a data source from Clarkson Research which reports what shipping brokers are willing to provide, otherwise charter rates will be withheld from public disclosure for confidential reasons (Cridland, 2010). As such, a large proportion of missing values (54%) impacts strongly on the sample size. In fact, Veenstra and van Dalen (2008) also highlighted that the available dataset cannot cover overall market activities.

Besides, the present study has not considered non-observable characteristics of charterers and owners and macroeconomic factors such as oil price, demand and supply for oil, vessel supply, and regulation which imply bargaining power and market conditions and in turn have a stronger impact on the variation of charter rates. At the same time, our study limits at the spot market without investing the relationship with the time charter market. The picture is thus still incomplete and should be investigated in the future study.

Last but not least, the encoding approach for categorical variables should also reconsider for the future research. With ability to deal with categorical predictors that contain a considerable number of levels, target encoding seems to be an appropriate and efficient method in this case. However, a disadvantage of target encoding is the possibility of data leakage as the method bases on the mean value of target variable in each category to measure the encoding value for that category, leading to overfitting. This issue can be tackled by using Leave-One-Out or K-fold validation, at the compensation of interpretability. Since the objective of this paper is to find a balance between accuracy and interpretability of an artificial intelligent model as XGBoost, target encoding is implemented without Leave-One-Out or K-Fold cross validation in order to maintain one encoding value for a category. We also acknowledge more intuitive encoding methods such that routes are translated to numerical values according to their distance or direction from one port (e.g., Persian Gulf), charterers and owners can be encoded by their capital or size. However, we were not able to implement those approaches due to time constraint. Although using only train data to generate the encoding values can decrease the possibility of data leakage to an extent, future study may develop a more intuitive and comprehensive approach to enhance the performance yet avoid overfitting.

References

Adland, R., Alger, H., Banyte, J., Jia, H. (2017a). *Does fuel efficiency pay? Empirical evidence from the dry bulk time charter market revisited.* Transport. Res. Part A 95, 1–12.

Adland, R., Cariou, P., Wolff, F.C. (2018). *Comparing transaction-based and expert-generated price indices in the market for offshore support vessels.* Working Papers halshs-01843720, HAL.

Adland, R., Cullinane, K. (2006). *The non-linear dynamics of spot freight rates in tanker markets.* Transp. Res. Part E 42, 211–224.

Adland, R., Jia, H. (2008). *Charter market default risk: A conceptual approach.* Transportation Research Part E: Logistics and Transportation Review, Volume 44, Issue 1, Pages 152-163,ISSN 1366-5545.

Adland, R., Jia, H., Lu, J. (2008). *Price dynamics in the market for Liquid Petroleum Gas transport.* Energy Econ. 30, 818–828.

Adland, R., P. Cariou, and F. C. Wolff (2017b). *What makes a freight market index? An empirical analysis of vessel fixtures in the offshore market.* Transportation Research Part E: Logistics and Transportation Review, 104, 150–64.

Agnolucci, P., Smith, T., Rehmatulla, N. (2014). *Energy efficiency and time charter rates: energy efficiency savings recovered by ship owners in the Panamax market.* Transp. Res. Part A 66, 173–184.

Alderton, P. (2004). *Reeds Sea Transport Operation and Economics.* 5th ed. London: Adlard Coles Nautical.

Alizadeh, A.H., Talley, W.K. (2011a). *Vessel and voyage determinants of tanker freight rates and contract times.* Transp. Policy 18, 665–675.

Alizadeh, A.H., Talley, W.K. (2011b). *Microeconomic determinants of dry bulk shipping freight rates and contract times.* Transportation 38, 561–579.

Berg-Andreassen, J.A. (1996). *Some properties of international maritime statistics.* Marit. Policy Manage. 23, 381–395.

Bjerksund, P., Ekern, S. (1995). *Contingent claims evaluation for mean-reverting cash flows in shipping.* In: Trigeorgis, L. (Ed.), Real Options in Capital Investments, Models, Strategies, and Applications. Praeger, Westport.

Brownlee, J. (2018). *Better Deep Learning: Traion Faster, Reduce Overfitting, and Make Better Predictions.* Machine Learning Mastery.

Christoph, M. (2019). *Interpretable machine learning. A Guide for Making Black Box Models Explainable.* Retrieved from https://christophm.github.io/interpretable-ml-book/.

Cridland C. (2010). *Challenges in measuring supply & demand.* In IMSF Annual Conference 2010.

Dwivedi, R. (2020). *Introduction to XGBoost Algorithm for Classification and Regression.* Retrieved from: https://www.analyticssteps.com/blogs/introduction-xgboost-algorithm-classification-and-regression

EIA (2017). *World Oil Transit Chokepoints.* Retrieved from: https://www.eia.gov/international/content/analysis/special_topics/World_Oil_Transit_Chokepoints/wotc.pdf

Euronav (2017). *The Basics of the Tanker Shipping Market.* Retrieved from: https://www.euronav.com/media/65361/special-report-2017-eng.pdf

Franses, P., Veenstra, A. (1997). *A co-integration approach to forecasting freight rates in the dry bulk shipping sector.* Transp. Res. 31, 447–458.

Friedman, Jerome H. (2001). *Greedy function approximation: A gradient boosting machine.* Annals of statistics: 1189-1232.

Hastie, T., Tibshirani, R., and Friedman, J. (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer New York.

Hyndman, R.J. and Athanasopoulos, G. (2013). *Forecasting: principles and practice.* OTexts.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning.* Springer Series in Statistics. Springer New York.

Kavussanos, M.G., Alizadeh, A.A., 2002. *The expectations hypothesis of the term structure and risk premia in dry bulk shipping freight markets.* Journal of Transport Economics and Policy 36 (2), 267–304.

Kavussanos, M.G., Alizadeh, A.H. (2001). *Seasonality patterns in dry bulk shipping spot and time charter freight rates.* Transport. Res. Part E: Logist. Transport. Rev. 37, 443–467.

Koehn, S. (2008). *Generalized Additive Models in the Context of Shipping Economics*, PhD thesis, University of Leicester.

Köhn, S., Thanopoulou, H. (2011). *A GAM assessment of quality premium in the dry bulk time charter market.* Transp. Res. Part E 47, 709–721.

Larsen, K. (2015). *GAM: The Predictive Modeling Silver Bullet.* Retrieved from: https://multithreaded.stitchfix.com/blog/2015/07/30/gam/

Laulajainen, R. (2007). *Dry bulk shipping market inefficiency, the wide perspective.* J. Transp. Geogr. 15(3), 217–224.

Lundberg, S. (2018). *Interpretable Machine Learning with XGBoost.* Retrieved from: https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27

Lundberg, S.M. and Lee, S.I. (2017). *A Unified Approach to Interpreting Model Predictions*. In Advances in Neural Information Processing Systems 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 4765–74. Curran Associates, Inc.

Morde, V. (2019). *XGBoost Algorithm: Long May She Reign!*. Retrieved from https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

Mozumdar, A. (2020). *A guide to encoding categorical features using R.* Retrieved from https://www.r-bloggers.com/2020/02/a-guide-to-encoding-categorical-features-using-r/#:~:text=Target%20encoding%20is%20also%20very,by%20specifying%20the%20sigma%20argument.

Olsen, M. and da Fonseca, T.R.K. (2017). *Investigating the predictive ability of AIS-data: the case of arabian gulf tanker rates.*

Parker, S. (2014). *Matching in the oil tanker industry: implications for energy efficiency*, PhD thesis, UCL (University College London).

Poblacion, J. (2015). *The stochastic seasonal behavior of freight rate dynamics.* Marit. Econ. Logist. 17, 142–162.

Población, J. (2017). *Are recent tanker freight rates stationary?* Marit. Econ. Logist. Kavussanos, M.G. (1996). Comparisons of volatility in the dry-cargo ship sector: Spot versus time charters, and smaller versus larger vessels. J. Transport Econ. Policy 30, 67–82.

Prochazka, V., Adland, R., Wolff, F.C. (2019), *Contracting decisions in the crude oil transportation market: Evidence from fixtures matched with AIS data.* Transportation Research Part A: Policy and Practice, Volume 130, Pages 37-53, ISSN 0965-8564.

Regli, F. (2019). *Essays on Crude Oil Tanker Markets.* PhD Series, No. 12.2019, ISBN 9788793744677, Copenhagen Business School (CBS), Frederiksberg.

Research and Market (2020). *Global Crude Tanker Market: Insights & Forecast with Potential Impact of COVID-19: 2020 Edition.* Retrieved from: https://www.researchandmarkets.com/reports/5145047/global-crude-tanker-market-insights-and-forecast?utm_source=BW&utm_medium=PressRelease&utm_code=6k5jbt&utm_campaign=1445193+-+Global+Crude+Tanker+(VLCC%2c+Suezmax+and+Aframax)+Market+Insights+%26+Forecast+Report+2020+with+Potential+Impact+of+COVID-19&utm_exec=chdo54prd

Rmileh, A.A. (2019). *The Multiple faces of 'Feature importance' in XGBoost.* Retrieved from https://towardsdatascience.com/be-careful-when-interpreting-your-features-importance-in-xgboost-6e16132588e7

Saxena, S. (2020). *Here's All you Need to Know About Encoding Categorical Data (with Python code).* Retrieved from https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/

Shah, T. (2017). *About Train, Validation and Test Sets in Machine Learning.* Retrieved from https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

Shapley, Lloyd S. (1953). *A Value for n-Person Games.* In Contributions to the Theory of Games Ii, edited by Harold W. Kuhn and Albert W. Tucker, 307–17. Princeton: Princeton University Press.

Shrivastava, S. (2014). *Cross Validation in Time Series.* Retrieved from: https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4

Stopford R. M., 2009. *Maritime Economics.* Routledge, London.

Tamvakis, M.N. (1995). *An investigation into the existence of a two-tier spot freight market for crude oil tankers.* Marit. Policy Manage. 22, 81–90.

Tamvakis, M.N., Thanopoulou, H.A. (2000). *Does quality pay? The case of the dry bulk market.* Transport. Res. Part E 36, 297–307.

Tvedt, J. (1996). *Market Structures, Freight Rates and Assets in Bulk Shipping.* Thesis (Ph.D.), Norwegian School of Economics and Business Administration, Bergen, Norway.

Tvedt, J. (1997). *Valuation of VLCCs under income uncertainty.* Marit. Policy Manage. 24, 159–174.

Tvedt, J. (2011). *Short-run freight rate formation in the VLCC market: A theoretical framework.* Maritime Economics & Logistics 13(4), 442–455.

Veenstra, A., van Dalen, J. (2008). *Price indices for ocean charter contracts.* In: The Second International Index Measures Congress, Washington, Digital proceedings.

Wood S. (2006). *Generalized Additive Models: An Introduction with R* (Chapman & Hall/CRC Texts in Statistical Science). 1st edition. Chapman and Hall/CRC.

Worldscale Association Limited (n.d.). *Definition of Worldscale.* Retrieved from https://www.worldscale.co.uk/help#t1

XGBoost Documentation (2019). *Xgboost parameters — xgboost 0.83.dev0 documentation.* Retrieved from https://xgboost.readthedocs.io/en/latest/parameter.html.

Appendix

A1. Overview of Quantitative Variables

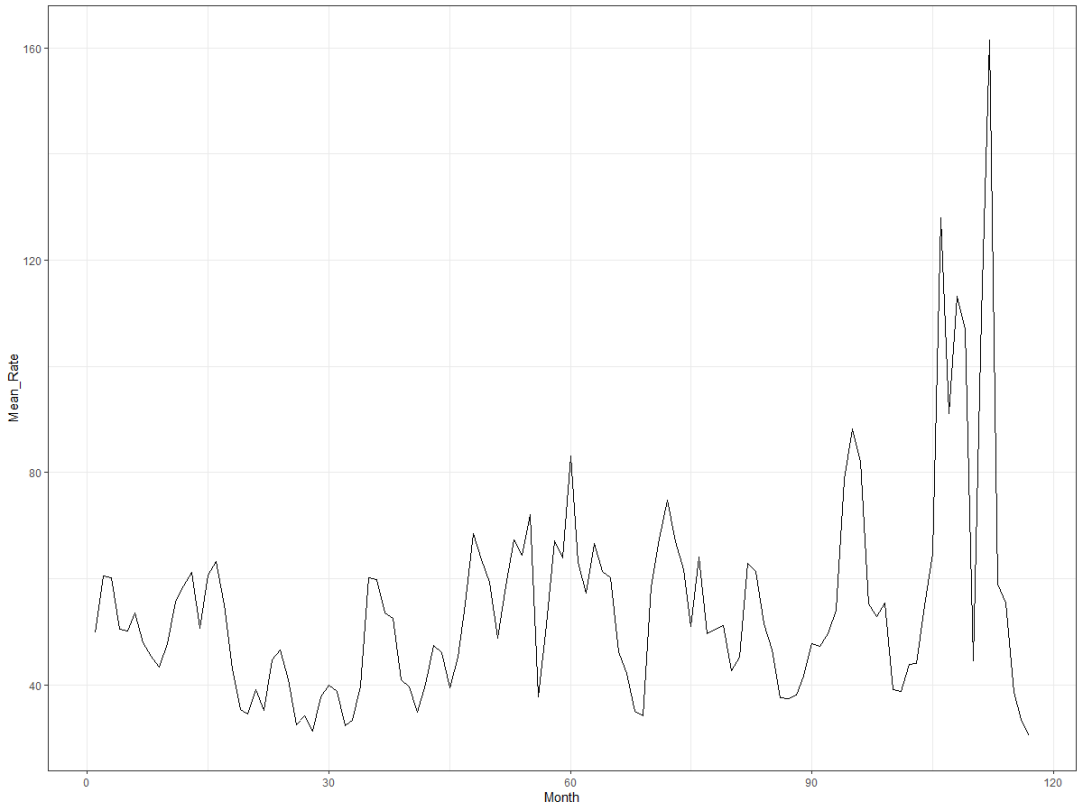


Figure A1.1 - Trends of freight rates over time (2011-2020)

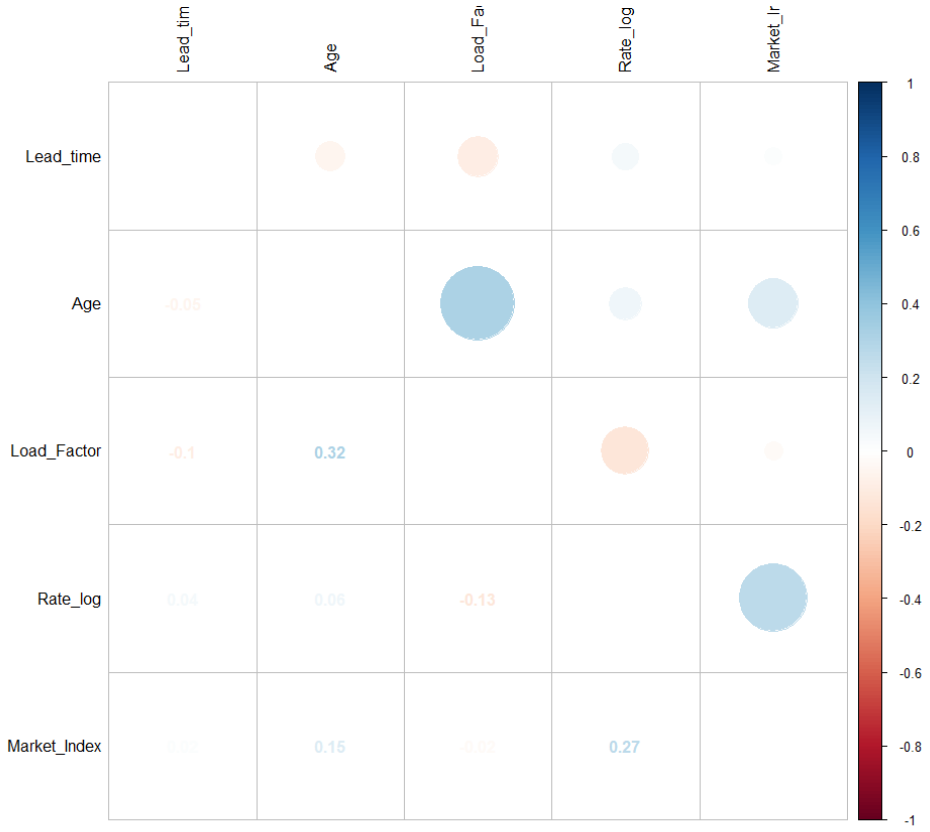


Figure A1.2 - Multicollinearity among quantitative variables

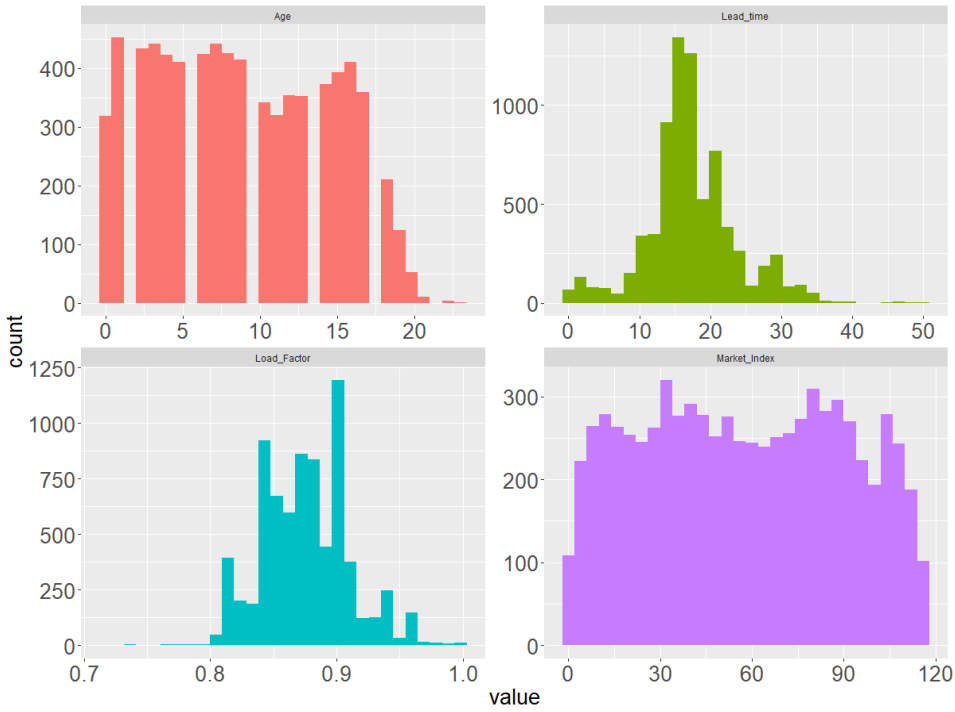


Figure A1.3 - Distribution of quantitative variables

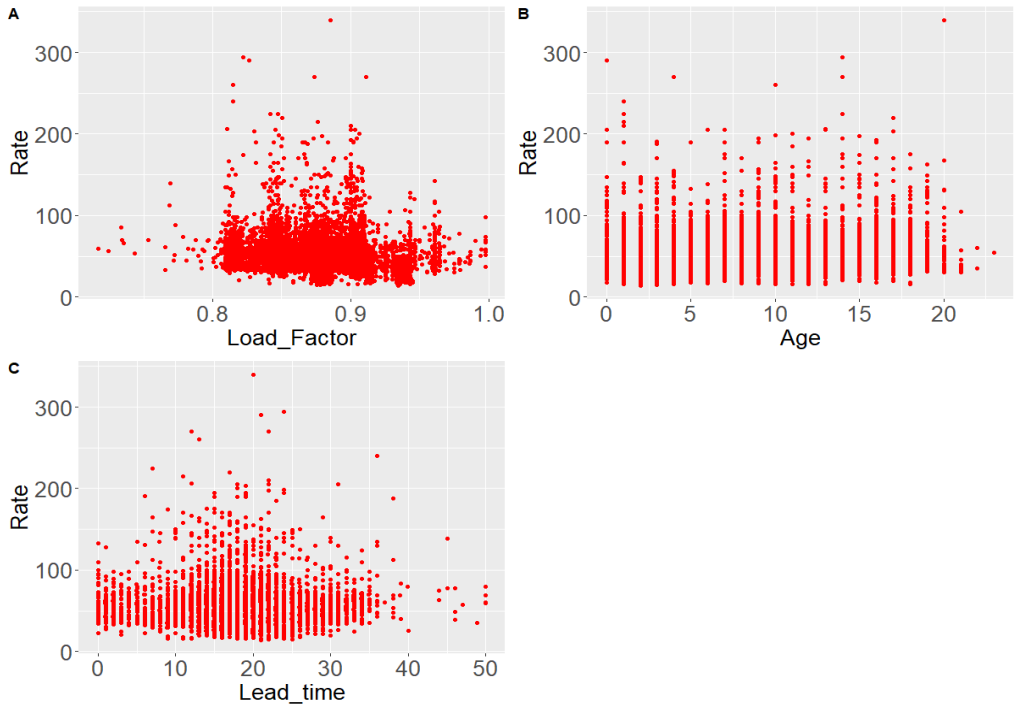


Figure A1.4 - The scatterplots between rates and quantitative dependent variables.

A2. Encoded Values and Original Categorical Values

No.	Route	Route encoded	Fixtures	Average Rate log	Average Rate	Distance (nm)
1	WEST AFRICA_UKC	4.136738526	26	4.14	62.88	4,310
2	WEST AFRICA_INDIA	4.086484153	16	4.09	59.12	7,188
3	**PERSIAN GULF_INDIA**	**4.058106354**	**825**	**4.06**	**59.69**	**1,352**
4	WEST AFRICA_EAST	4.051573787	123	4.05	57.81	4,883
5	OTHER	4.047129936	32	4.05	57.88	
6	**PERSIAN GULF_CHINA**	**3.976580301**	**1155**	**3.98**	**54.29**	**5,852**
7	WEST AFRICA_USG	3.967129978	45	3.97	52.77	5,749
8	**WEST AFRICA_CHINA**	**3.966587016**	**692**	**3.97**	**53.21**	**9,579**
9	WEST AFRICA_TAIWAN	3.964143956	64	3.96	52.64	9,118
10	**PERSIAN GULF_JAPAN**	**3.961210515**	**320**	**3.96**	**53.06**	**6,358**
11	**PERSIAN GULF_EAST**	**3.949116168**	**278**	**3.95**	**52.48**	**11,765**
12	**PERSIAN GULF_THAILAND**	**3.93523202**	**382**	**3.94**	**52.02**	**4,409**
13	**PERSIAN GULF_SINGAPORE**	**3.921566214**	**293**	**3.92**	**51.38**	**2,435**
14	PERSIAN GULF_MALAYSIA	3.914099844	72	3.91	50.78	3,671
15	**PERSIAN GULF_TAIWAN**	**3.898148755**	**319**	**3.9**	**50.05**	**5,290**
16	PERSIAN GULF_SOUTH AFRICA	3.881657767	57	3.88	49.19	4,514
17	**PERSIAN GULF_SOUTH KOREA**	**3.868138701**	**1079**	**3.87**	**48.36**	**6,187**
18	PERSIAN GULF_UKC	3.510667215	82	3.51	33.84	6,360
19	PERSIAN GULF_CANADA	3.457132813	59	3.46	32.2	11,353
20	**PERSIAN GULF_USG**	**3.419672815**	**384**	**3.42**	**30.61**	**13,436**
21	PERSIAN GULF_WEST	3.312111738	64	3.31	27.66	14,236

Note: Top 10 most active routes are highlighted.

Table A2.1 - Route encoded values vs Original categorical values (Train set). Source: Authors' calculations, distance data retrieved from: Stopford (2009) and Parker (2014).

No	Route	Route encoded	Fixtures	Average Rate log	Average Rate	Distance (nm)
1	WEST AFRICA_UKC	4.136738526	4	4.25	70.88	4,310
2	WEST AFRICA_INDIA	4.086484153	2	3.57	34.5	7,188
3	**PERSIAN GULF_INDIA**	**4.058106354**	**159**	**4.36**	**91.79**	**1,352**
4	WEST AFRICA_EAST	4.051573787	69	4.12	66.22	4,883
5	OTHER	4.047129936	4	4.29	79.75	
6	**PERSIAN GULF_CHINA**	**3.976580301**	**348**	**4.22**	**76.03**	**5,852**
7	WEST AFRICA_CHINA	3.966587016	120	4.19	71.21	9,579
8	WEST AFRICA_TAIWAN	3.964143956	3	4.16	81.33	9,118
9	**PERSIAN GULF_JAPAN**	**3.961210515**	**21**	**4.23**	**79.77**	**6,358**
10	**PERSIAN GULF_EAST**	**3.949116168**	**70**	**4.1**	**67.61**	**11,765**
11	**PERSIAN GULF_THAILAND**	**3.93523202**	**24**	**4.11**	**67.96**	**4,409**
12	**PERSIAN GULF_SINGAPORE**	**3.921566214**	**22**	**4.02**	**60.26**	**2,435**
13	PERSIAN GULF_MALAYSIA	3.914099844	9	4.1	66.25	3,671
14	**PERSIAN GULF_TAIWAN**	**3.898148755**	**77**	**4.05**	**64.58**	**5,290**
15	PERSIAN GULF_SOUTH AFRICA	3.881657767	5	3.99	57.8	4,514
16	**PERSIAN GULF_SOUTH KOREA**	**3.868138701**	**164**	**4.03**	**61.36**	**6,187**
17	PERSIAN GULF_UKC	3.510667215	2	3.44	33.25	6,360
18	PERSIAN GULF_CANADA	3.457132813	2	4.14	107	11,353
19	**PERSIAN GULF_USG**	**3.419672815**	**12**	**3.37**	**31.17**	**13,436**
20	PERSIAN GULF_WEST	3.312111738	1	3.33	27	14,236

Note: Top 10 most active routes are highlighted.

Table A2.2 - Route encoded values vs Original categorical values (Test set). Source: Authors' calculations, distance data retrieved from: Stopford (2009) and Parker (2014).

No.	Charterer	Charterer_encoded	Fixtures	Average_Rate_log	Average_Rate
1	PETCO	4.33073334	1	4.33	75
2	PLAINS	4.317399217	2	4.32	74
3	TAIYO	4.25561271	1	4.26	69.5
4	CLEARLAKE	4.252661509	2	4.25	70
5	BORL	4.229191301	20	4.23	72.15
6	NITC	4.183243287	2	4.18	64.75
7	UML	4.143134726	1	4.14	62
8	SOCAR	4.141302041	3	4.14	66.67
9	CONOCOPHILLIPS	4.13176787	3	4.13	61.33
10	OCCIDENTAL	4.130698654	2	4.13	65.25
11	CHINESE	4.122148668	6	4.12	62.25
12	GLENCORE/ALPINE	4.110873864	1	4.11	60
13	KANGQI	4.103370276	9	4.1	59.97
14	**IOC**	**4.097693778**	**429**	**4.1**	**61.95**
15	HPCL	4.079163629	5	4.08	62.9
16	PETRODIAMOND	4.077537444	1	4.08	58
17	JAPANESE	4.06066591	2	4.06	57.25
18	RESOURCE MARINE	4.059860384	6	4.06	59.54
19	PETROINEOS	4.059048399	35	4.06	58.5
20	MITSUI	4.038413016	2	4.04	57.75
21	EQUINOR	4.032319846	29	4.03	57.41
22	ST SHIPPING	4.032140686	16	4.03	59.95
23	COSMO	4.031756196	17	4.03	57.09
24	BPCL	4.027514181	41	4.03	57.23
25	ADNATCO	4.026561341	7	4.03	57.61
26	CNR	4.025835686	43	4.03	57.34
27	PETROCHINA	4.025351691	1	4.03	55
28	FUJI OIL	4.017682759	8	4.02	55.62
29	HMEL	4.012842427	55	4.01	57.19
30	JX	4.006274657	3	4.01	54.5
31	CHEM CHINA	4.005495589	30	4.01	55.23
32	TRAFIGURA	4.0052453	20	4.01	55.48
33	ESSAR	4.001297104	28	4	55.62
34	SHOWA SHELL	3.998200702	1	4	53.5
35	ZHUHAI ZHENRONG	3.997980252	11	4	54.09
36	**RELIANCE**	**3.995422174**	**247**	**4**	**55.96**
37	SPC	3.989251434	113	3.99	54.87
38	**DAYHARVEST**	**3.982862017**	**310**	**3.98**	**54.45**
39	SK ENERGY	3.973054139	73	3.97	53.87
40	**UNIPEC**	**3.97127567**	**1062**	**3.97**	**53.86**
41	SAROIL	3.970291914	1	3.97	52
42	IDEMITSU	3.963062713	91	3.96	53.55
43	NPI	3.962043826	77	3.96	53.31
44	SCI	3.96081317	1	3.96	51.5
45	MITSUBISHI	3.956530383	82	3.96	52.51
46	**PTT**	**3.950519382**	**329**	**3.95**	**52.87**
47	GLASFORD	3.944303897	158	3.94	52.43
48	PHILLIPS	3.931825633	1	3.93	50
49	CONOCO	3.92951723	21	3.93	50.48
50	SASOL	3.927788244	3	3.93	51.83
51	CNOOC	3.926526264	116	3.93	51.32
52	STATOILHYDRO ASA	3.921973336	1	3.92	49.5
53	HYUNDAI OILBANK	3.920350573	98	3.92	51.07
54	TESORO	3.91934467	3	3.92	51.83
55	FORMOSA	3.915038348	138	3.92	50.68
56	PETRONAS	3.898666434	48	3.9	50.58
57	**CPC**	**3.893169764**	**218**	**3.89**	**49.67**
58	CLEARLAKE SHPG	3.891548992	8	3.89	49
59	PETROBRAS	3.889942558	2	3.89	48
60	MERCURIA	3.888876828	42	3.89	49.54
61	SAHARA	3.881563798	1	3.88	47.5
62	TONEN	3.879705559	20	3.88	49.34
63	SSANGYONG	3.876395828	1	3.88	47.25
64	**CHEVTEX**	**3.874343034**	**427**	**3.87**	**48.9**
65	SK CORP	3.870873623	78	3.87	48.39
66	IRPC	3.868467662	55	3.87	48.09
67	HMM	3.866110131	62	3.87	47.85
68	ENGEN	3.864106762	12	3.86	48.21
69	S-OIL	3.862665019	2	3.86	47.88
70	CHEMOIL	3.850147602	1	3.85	46
71	**S.OIL**	**3.834709681**	**382**	**3.83**	**46.68**
72	**SHELL**	**3.82621748**	**286**	**3.83**	**47.52**
73	BP AMOCO	3.821104107	89	3.82	46.93
74	TOTAL	3.811332906	149	3.81	47.08
75	BRIGHTOIL	3.777898001	10	3.78	43.58
76	SEARIVER	3.745251268	81	3.75	43.88
77	ENI	3.702699535	15	3.7	42.1
78	VITOL	3.654773491	2	3.65	38.75
79	BAHRI	3.653097899	23	3.65	38.85
80	VALERO	3.64790586	5	3.65	37.6
81	VELA	3.645782334	6	3.65	37.33
82	**EXXONMOBIL**	**3.645435487**	**266**	**3.65**	**39.61**
83	LITASCO	3.630118632	48	3.63	39.26
84	CEPSA	3.583518938	1	3.58	35
85	TOTAL/FINA/ELF	3.571763756	62	3.57	36.9
86	KPC	3.566231935	78	3.57	37.27
87	STATOIL	3.551449913	15	3.55	36.87
88	ENOC	3.534298902	2	3.53	33.75
89	PHILLIPS 66	3.523028183	16	3.52	34.89
90	KOCH	3.512681309	20	3.51	33.9
91	IRVING	3.468854899	57	3.47	32.57
92	PBF ENERGY	3.401197382	2	3.4	29
93	LUKOIL	3.349904087	1	3.35	27.5
94	PBF	3.300442908	5	3.3	27.8
95	MAR/ASHLAND	2.917770732	1	2.92	17.5

Note: Top 10 most active charterers are highlighted.

Table A2.3 - Charterer encoded values vs Original categorical values (Train set).

No.	Charterer	Charterer_encoded	Fixtures	Average_Rate_log	Average_Rate
1	BORL	4.229191301	6	4.75	131.75
2	UML	4.143134726	1	3.93	50
3	CHINESE	4.122148668	3	4.92	139.17
4	**IOC**	**4.097693778**	**80**	**4.31**	**87.36**
5	HPCL	4.079163629	1	4.09	58.5
6	PETROINEOS	4.059048399	8	4.34	90.84
7	EQUINOR	4.032319846	15	4.2	71.57
8	ST SHIPPING	4.032140686	13	4.12	65.4
9	COSMO	4.031756196	4	4.11	64.56
10	BPCL	4.027514181	26	4.15	73.63
11	CNR	4.025835686	26	4.44	94.38
12	HMEL	4.012842427	3	4.2	65.67
13	JX	4.006274657	2	3.76	42.25
14	CHEM CHINA	4.005495589	7	3.96	57.93
15	TRAFIGURA	4.0052453	6	4.02	59.62
16	**RELIANCE**	**3.995422174**	**24**	**4.38**	**90.21**
17	SPC	3.989251434	19	4.24	78.84
18	**DAYHARVEST**	**3.982862017**	**68**	**4.22**	**75.64**
19	SK ENERGY	3.973054139	35	3.95	57.94
20	**UNIPEC**	**3.97127567**	**278**	**4.18**	**71.16**
21	IDEMITSU	3.963062713	7	4.39	89.43
22	NPI	3.962043826	7	4.4	85.71
23	SCI	3.96081317	1	4.2	66
24	**PTT**	**3.950519382**	**25**	**4.12**	**68.42**
25	GLASFORD	3.944303897	21	4.27	77.74
26	CNOOC	3.926526264	31	4.13	69.29
27	HYUNDAI OILBA	3.920350573	26	4.06	63.24
28	FORMOSA	3.915038348	44	4.13	70.98
29	ADNOC	3.906671587	1	3.76	42
30	ATC	3.906671587	1	4.11	60
31	HANWHA	3.906671587	1	4.78	118.5
32	HANWHA - TOT	3.906671587	1	4.19	65
33	HENGYI	3.906671587	6	4.88	142.5
34	MRPL	3.906671587	14	4.83	142.74
35	NSRP	3.906671587	1	3.93	49.75
36	RONGSHENG	3.906671587	14	3.98	60.57
37	PETRONAS	3.898666434	9	4.1	66.25
38	**CPC**	**3.893169764**	**34**	**4.01**	**61.95**
39	MERCURIA	3.888876828	3	3.96	55.67
40	SAHARA	3.881563798	2	3.57	34.5
41	**CHEVTEX**	**3.874343034**	**48**	**4.04**	**59.03**
42	S-OIL	3.862665019	6	3.81	51.75
43	**S.OIL**	**3.834709681**	**33**	**3.99**	**61.89**
44	**SHELL**	**3.82621748**	**40**	**4.02**	**60.56**
45	BP AMOCO	3.821104107	13	4.06	62.5
46	TOTAL	3.811332906	44	4.17	73.77
47	SEARIVER	3.745251268	2	4.16	68.5
48	ENI	3.702699535	4	3.93	51.31
49	**EXXONMOBIL**	**3.645435487**	**14**	**3.67**	**42.48**
50	LITASCO	3.630118632	3	3.93	50.83
51	KPC	3.566231935	16	3.97	63.11
52	KOCH	3.512681309	19	4.11	68.43
53	IRVING	3.468854899	2	4.14	107

Note: Top 10 most active charterers are highlighted.

Table A2.4 - Charterer encoded values vs Original categorical values (Test set)

No.	Owner	Owner_encoded	Fixtures	Average_Rate_log	Average_Rate
1	Hellenic Tankers	4.454347296	1	4.45	85
2	Kmarin	4.436751534	1	4.44	83.5
3	TOP-NYK MarineOne	4.30661519	2	4.31	74
4	ADS Crude	4.264857236	2	4.26	71
5	JX Ocean	4.262679877	1	4.26	70
6	PT Sukses Osean	4.262679877	1	4.26	70
7	HI Gold Ocean No.11	4.253904945	3	4.25	69.5
8	Dannebrog Invest	4.248495242	1	4.25	69
9	Dryships	4.237560207	2	4.24	71.5
10	Sentek Marine	4.227493574	3	4.23	68.33
11	Eastern Med Mar	4.21510604	18	4.22	68.28
12	Agritrade Resources	4.196808823	3	4.2	65.5
13	DS Schiffahrt GmbH	4.194682581	7	4.19	69.29
14	Jinhai Heavy Ind	4.151039906	1	4.15	62.5
15	Seven Islands Shpg	4.14117683	7	4.14	63.25
16	Sea Fortune Tankers	4.127134385	1	4.13	61
17	Bumi Armada Nav	4.123741002	2	4.12	63.5
18	Oaktree Capital	4.123370008	18	4.12	62.81
19	Cardiff Marine	4.119605605	13	4.12	63.4
20	TMM Co Ltd	4.110873864	1	4.11	60
21	CSG Tanker Shanghai	4.106349797	6	4.11	60.33
22	Far East Leasing	4.100206998	4	4.1	59.88
23	**Dynacom Tankers Mgm**	**4.100063707**	**116**	**4.1**	**62.85**
24	Delta Tankers	4.099473201	7	4.1	60.18
25	Korea Line	4.094344562	1	4.09	59
26	Mitsui & Co. Ltd.	4.094344562	1	4.09	59
27	SBM Offshore	4.089459666	2	4.09	60.75
28	TI Guardian K/S	4.077537444	1	4.08	58
29	CSG Tanker Dalian	4.074644583	11	4.07	59.68
30	Formosa Plastics Co	4.065425218	31	4.07	58.86
31	**Ocean Tankers**	**4.065293986**	**135**	**4.07**	**58.86**
32	Andriaki Shipping	4.064580862	9	4.06	58.42
33	Showa Shell Sekiyu	4.060443011	1	4.06	57
34	Sinochem Group	4.051139618	7	4.05	56.96
35	**New Shipping**	**4.049316092**	**181**	**4.05**	**58.85**
36	Awilco AS	4.048975674	23	4.05	58.2
37	Nat. Iranian Tanker	4.043051268	1	4.04	56
38	SK Shipping	4.03972375	99	4.04	58.03
39	Minerva Marine	4.039476087	11	4.04	56.89
40	Maersk Tankers	4.035599571	7	4.04	56.79
41	COSCO Group	4.034125797	4	4.03	56
42	New Shipping Ltd.	4.032966912	9	4.03	56
43	CV Shipping	4.025351691	1	4.03	55
44	Landbridge	4.022831808	12	4.02	57.12
45	Saga Tankers	4.02041067	9	4.02	55.67
46	TMT Co. Ltd.	4.007115187	40	4.01	55.05
47	Polembros Shipping	3.995168012	84	4	56.19
48	AET Tankers	3.993269903	31	3.99	57.4
49	Polembros Shpg.	3.988883044	32	3.99	53.72
50	Knightsbridge Tank.	3.98635894	13	3.99	54.08
51	Frontline Ltd.	3.986256273	19	3.99	54.47
52	Global Energy M'time	3.985697486	5	3.99	54.05
53	Thenamaris	3.985204514	34	3.99	54.28
54	Intl Seaways	3.982358613	21	3.98	56.1
55	**Maran Tankers Mgmt**	**3.976811386**	**146**	**3.98**	**55.21**
56	Marcare Shpg. Co.	3.975122062	3	3.98	52.33
57	Zodiac Maritime	3.974324212	4	3.97	54.75
58	**Dynacom Tankers Mng**	**3.971801781**	**165**	**3.97**	**54.83**
59	Eastern Med	3.964690905	65	3.96	53.28
60	Essar Shipping	3.961971769	14	3.96	52.5
61	Gener8 Maritime	3.961126518	63	3.96	55.1
62	Int. Energy Trans.	3.955769917	7	3.96	51.71
63	DS Tankers	3.955509272	65	3.96	53.83
64	Oman Shipping Co	3.955138406	34	3.96	54.12
65	**Aeolos Management**	**3.953054341**	**162**	**3.95**	**53.19**
66	SBM Production Cont.	3.951243719	1	3.95	51
67	Rimau Shipping	3.950133018	2	3.95	53.5
68	NGM Energy	3.94911495	23	3.95	53.66
69	Pantheon Tankers	3.947967158	78	3.95	53.79
70	Centrofin Mngt.	3.946371871	14	3.95	52.34
71	Euronav/Sincere JV	3.945197696	2	3.95	52.75
72	Mitsui O.S.K. Lines	3.942995909	127	3.94	52.34
73	**Shpg Corp of India**	**3.942877641**	**122**	**3.94**	**52.59**
74	**DHT Management**	**3.942085646**	**167**	**3.94**	**53.53**
75	Kyoei Tanker Co.	3.94087057	9	3.94	51.33
76	NGM Energy S.A.	3.939898951	11	3.94	53.95
77	Kuwait Oil Tanker	3.938784344	44	3.94	52.23
78	Nat Iranian Tanker	3.934159478	19	3.93	51.18
79	Sinokor Merchant	3.934007704	47	3.93	52.14
80	Olympic Shpg & Mgmt	3.933525674	63	3.93	52.77
81	Dr. Peters	3.932310803	31	3.93	51.44
82	Centrofin Mgmt	3.931854185	17	3.93	51.68
83	U-Ming Marine Trans	3.931056106	2	3.93	50
84	Mitsui OSK Lines	3.931045185	35	3.93	52.91
85	Smart Tankers	3.930285244	17	3.93	51.6
86	Enesel SA	3.929729847	16	3.93	50.89
87	K-Line	3.927409248	40	3.93	51.18
88	Navios Midstream	3.926492735	31	3.93	51.83
89	Ship Finance Intl	3.92498436	86	3.92	51.63
90	Tai Chong Cheang	3.923717134	12	3.92	50.62
91	China VLCC	3.921068785	73	3.92	54.86
92	Srithai Marine Corp	3.919122365	2	3.92	53
93	U-Ming Marine Tran.	3.918913754	13	3.92	49.88
94	**Euronav NV**	**3.9176748**	**191**	**3.92**	**51.86**
95	Tsakos Shpg. & Trad.	3.917425571	12	3.92	50.54
96	Hermes Marine Mgmt	3.916431074	4	3.92	58.38
97	Ridgebury Tankers	3.915259386	33	3.92	52.01
98	Aramo Shipping	3.914508313	42	3.91	50.86
99	Teekay Tankers	3.912760356	28	3.91	50.31
100	Nissen Kaiun	3.911822965	2	3.91	49
101	Windsor Petroleum	3.908937698	19	3.91	50.78
102	Gulf Marine Mngt.	3.908161807	19	3.91	50.95
103	Minerva Marine Inc.	3.907980435	14	3.91	49.96
104	Formosa Plastics Co.	3.905171362	37	3.91	50.51
105	Thenamaris (Mgmt.)	3.904120353	7	3.9	50.71
106	BW Offshore	3.90197267	1	3.9	48.5
107	MODEC	3.901264576	3	3.9	49.25
108	Capital Ship Mngt	3.901147689	10	3.9	52.92
109	Eastern Pacific Shpg	3.900849074	2	3.9	50
110	Sinokor Merchant Mar	3.900489116	6	3.9	48.83
111	Tanker Pacific Mngt.	3.899299275	21	3.9	49.27
112	CSC Oil Trans.	3.899006816	3	3.9	49.33

No.	Owner	Owner_encoded	Fixtures	Average_Rate_log	Average_Rate
113	Shpg. Corp. of India	3.898897971	44	3.9	49.27
114	Atlas Maritime Ltd.	3.898470198	4	3.9	50
115	NS United K.K.	3.898030097	5	3.9	48.4
116	Bahri	3.89759426	81	3.9	49.57
117	Frontline 2012	3.896518105	39	3.9	50.97
118	Brightoil Petroleum	3.893118559	67	3.89	50.23
119	Almi Tankers	3.892975691	3	3.89	54.33
120	GMS Inc	3.892759185	8	3.89	48.94
121	CSET Tanker Dalian	3.892432668	41	3.89	52.31
122	HMM	3.892326975	22	3.89	50.03
123	China Shpg. Tankers	3.891103135	45	3.89	48.99
124	Olympic Shipping	3.890799388	38	3.89	49.48
125	Frontline	3.888961286	71	3.89	51.01
126	Oriental Shipping	3.886526994	3	3.89	48.58
127	Delta Tankers Ltd.	3.885660853	19	3.89	48.91
128	BW Maritime	3.883617603	87	3.88	49.97
129	Sonatrach Petroleum	3.883572196	25	3.88	51.28
130	MISC	3.882625946	34	3.88	49.68
131	Nippon Yusen Kaisha	3.882451019	73	3.88	49.02
132	Hermes Marine Mngt	3.881510655	2	3.88	48
133	Ocean Tankers Pte	3.879690745	98	3.88	48.63
134	Navios Acquisition	3.879654588	113	3.88	49.46
135	Altomare SA	3.87794125	69	3.88	49.03
136	BW Maritime Pte.	3.8776362	51	3.88	48.27
137	Tsakos Energy Nav.	3.876737226	16	3.88	47.84
138	A. P. Moller	3.872895184	47	3.87	48.45
139	Unknown	3.871608426	5	3.87	47.4
140	Hyundai M.M.	3.866995997	20	3.87	47.95
141	P.T. Sukses Osean	3.866199189	10	3.87	47.12
142	Andriaki Shpg. Co.	3.864392162	26	3.86	48.07
143	Oak Maritime Canada	3.86325658	6	3.86	48
144	TMM Co. Ltd.	3.86171922	15	3.86	47.53
145	Iino Kaiun Kaisha	3.860308448	6	3.86	47.08
146	Enesel S.A.	3.859313224	28	3.86	48.15
147	DHT Management AS	3.857610967	8	3.86	47.12
148	Crude Carriers	3.855412738	13	3.86	47.23
149	HOSCO	3.85389132	39	3.85	47.23
150	Eurotankers	3.853383052	8	3.85	46.75
151	Idemitsu Kosan	3.852390077	6	3.85	47.17
152	Livanos Group	3.849268532	29	3.85	47.17
153	China Merchants Shpg	3.847150714	37	3.85	47.46
154	CSET Tanker Shanghai	3.846604412	32	3.85	47.91
155	Eastern Med. Mar.	3.844706373	81	3.84	47.13
156	Athenian Carriers	3.844253622	51	3.84	47.86
157	Nissen Kaiun K.K.	3.843613732	9	3.84	46.94
158	Altomare S.A.	3.843551683	65	3.84	47.08
159	Chandris (Hellas)	3.840283139	53	3.84	48.21
160	Hanjin Shipping	3.836543958	8	3.84	47.34
161	Navig8 Shipmgmt	3.835946412	6	3.84	47.08
162	Ship Finance Inter.	3.835718142	104	3.84	46.94
163	Alpha Tankers & Frt.	3.835338308	19	3.84	48
164	General Maritime	3.832930132	78	3.83	46.87
165	**Maran Tankers Mngt.**	**3.832802925**	**193**	**3.83**	**47.7**
166	Nereus Shipping	3.832787152	48	3.83	47.01
167	Cido Shipping	3.830967831	27	3.83	47.13
168	Nan Fung Shipping	3.830144809	3	3.83	45.17
169	Navig8 Shipmngt	3.828641396	1	3.83	45
170	Nissho Shpg. Co. Ltd	3.828641396	1	3.83	45
171	Sea Fortune Shipping	3.828641396	1	3.83	45
172	Athenian Tankers	3.827834135	53	3.83	47.5
173	Cardiff Marine Inc.	3.818797873	45	3.82	45.78
174	TRF	3.815082005	3	3.82	50
175	Wah Kwong Shpg	3.810298515	25	3.81	46.4
176	Tsakos Energy Nav	3.808020694	6	3.81	49
177	Titan Ocean	3.80666249	1	3.81	44
178	Kyoei Tanker	3.803127118	3	3.8	44.25
179	SK Shipping Co. Ltd.	3.7980313	110	3.8	44.85
180	Samos Steamship	3.788049031	14	3.79	44.5
181	Mercator	3.787777381	20	3.79	45.86
182	Euronav (UK) Agen.	3.785910247	13	3.79	44.31
183	TMT Co Ltd	3.777883836	8	3.78	43.22
184	ITCL	3.776077921	26	3.78	43.38
185	Marine Trust Ltd.	3.775624026	12	3.78	43.98
186	Reederei Nord	3.773008813	8	3.77	45.72
187	Metrostar Management	3.772760938	1	3.77	42.5
188	AET Tanker	3.771552295	44	3.77	44.34
189	Overseas Shipholding	3.763292794	55	3.76	44.04
190	Oman Shipping Co.	3.762872106	85	3.76	44.69
191	Neda Maritime Agency	3.759839922	55	3.76	45.05
192	Salamon AG	3.756550188	4	3.76	42.25
193	Phoenix Tankers	3.754276393	12	3.75	43.48
194	DK Maritime	3.736703838	41	3.74	42.27
195	Intl Energy Trans	3.728660366	2	3.73	42.25
196	Gulf Marine Deutsch	3.728254512	21	3.73	43.1
197	Idemitsu Tanker Co.	3.727315764	4	3.73	40.75
198	COSCO Dalian	3.725443685	12	3.73	41.75
199	Samco Shipholding	3.720623059	14	3.72	40.73
200	JX Tanker Co., Ltd.	3.713572067	1	3.71	40
201	Hanjin Shpg Co.	3.712729459	3	3.71	43.33
202	Foresight	3.699037448	4	3.7	39.75
203	Gulf Navigation	3.697054205	10	3.7	41.48
204	GC Tankers	3.694700061	29	3.69	43.63
205	CSC Nanjing Tanker	3.687967787	26	3.69	42.29
206	Capital Ship Mgmt	3.680376848	21	3.68	42.83
207	Singha Tanker Co	3.676300672	1	3.68	38.5
208	STASCO (Shell)	3.656943416	2	3.66	37.75
209	KMARIN	3.641724114	2	3.64	40
210	NS United KK	3.632015071	2	3.63	37
211	Tufton Oceanic	3.629195485	3	3.63	41.5
212	Capital Ship Mngt.	3.620127283	8	3.62	39.12
213	Sincere Navigation	3.613926037	18	3.61	40.24
214	Idemitsu Tanker	3.583518938	1	3.58	35
215	Meiji Shipping Co.	3.583518938	1	3.58	35
216	Wah Kwong Shpg.	3.581124096	3	3.58	37
217	A.P. Moller	3.555322239	29	3.56	36.14
218	Meiji Shipping Co	3.551322282	8	3.55	36.06
219	Russian Titan Shpg.	3.526360525	1	3.53	33
220	Mitsui & Co	3.496507561	1	3.5	32
221	Euronav UK Agencies	3.47613746	5	3.48	32.7
222	Sambouk Shipping	3.449987546	1	3.45	30.5
223	Clipper Group	3.391826136	3	3.39	31.17
224	GC Tankers Pte. Ltd.	3.295836866	1	3.3	26
225	Nord, Reederei, Ltd.	3.295836866	1	3.3	26
226	Sinotrans Shpg. Ltd.	3.258096538	1	3.26	25
227	CV Shipping Pte	3.135494216	1	3.14	22

Note: Top 10 most active owners are highlighted.

Table A2.5 - Owner encoded values vs Original categorical values (Train set)

No.	Owner	Owner encoded	Fixtures	Average Rate log	Average Rate
1	Hellenic Tankers	4.454347296	6	4.44	98.04
2	ADS Crude	4.264857236	12	4.13	68.29
3	JX Ocean	4.262679877	1	3.97	52
4	Sentek Marine	4.227493574	2	4.51	106.75
5	Agritrade Resources	4.196808823	2	3.9	48.5
6	Seven Islands Shpg	4.14117683	4	5.03	175.25
7	Cardiff Marine	4.119605605	2	3.77	42.5
8	**Dynacom Tankers Mgmt**	**4.100063707**	**57**	**4.29**	**82.86**
9	Delta Tankers	4.099473201	7	4.43	94.86
10	Korea Line	4.094344562	1	4.11	60
11	SBM Offshore	4.089459666	2	3.71	40
12	Formosa Plastics Co	4.065425218	11	4.05	60.07
13	**Ocean Tankers**	**4.065293986**	**25**	**4.17**	**67.59**
14	Andriaki Shipping	4.064580862	10	4.1	63.35
15	Sinochem Group	4.051139618	1	4.22	67
16	**New Shipping**	**4.049316092**	**58**	**4.12**	**67.21**
17	Awilco AS	4.048975674	15	4.32	83.7
18	SK Shipping	4.03972375	22	3.99	56.91
19	Minerva Marine	4.039476087	7	4.15	65.79
20	Landbridge	4.022831808	9	4.22	78.97
21	Polembros Shipping	3.995168012	3	4.3	81.83
22	AET Tankers	3.993269903	32	4.01	61.71
23	Global Energy M'time	3.985697486	1	3.33	27
24	Thenamaris	3.985204514	23	4.2	76.71
25	Intl Seaways	3.982358613	20	4.25	80.44
26	**Maran Tankers Mgmt**	**3.976811386**	**78**	**4.19**	**73.12**
27	Zodiac Maritime	3.974324212	13	4.31	84.12
28	Eastern Med	3.964690905	18	4.47	99.11
29	Essar Shipping	3.961971769	11	4.01	61.55
30	Oman Shipping Co	3.955138406	22	4.17	71.62
31	**Aeolos Management**	**3.953054341**	**34**	**4.26**	**80.12**
32	NGM Energy	3.94911495	4	4.37	96.75
33	Pantheon Tankers	3.947967158	27	4.07	67.3
34	**Shpg Corp of India**	**3.942877641**	**26**	**4.3**	**83.1**
35	**DHT Management**	**3.942085646**	**50**	**4.07**	**64.75**
36	Kuwait Oil Tanker	3.938784344	15	3.92	51.13
37	Sinokor Merchant	3.934007704	12	4.11	65.21
38	Olympic Shpg & Mgmt	3.933525674	31	4.07	68.1
39	Mitsui OSK Lines	3.931045185	12	3.92	51.85
40	Smart Tankers	3.930285244	9	4.17	79.28
41	Enesel SA	3.929729847	6	4.54	111
42	K-Line	3.927409248	9	3.89	53.61
43	Tai Chong Cheang	3.923717134	5	4.06	57.55
44	China VLCC	3.921068785	15	4.17	80.57
45	**Euronav NV**	**3.9176748**	**48**	**4.11**	**65.28**
46	Hermes Marine Mgmt	3.916431074	5	4.27	74.3

No.	Owner	Owner encoded	Fixtures	Average Rate log	Average Rate
47	Ridgebury Tankers	3.915259386	11	3.89	53.2
48	Aramo Shipping	3.914508313	4	4.09	62.44
49	Nissen Kaiun	3.911822965	2	4.42	90
50	Al-Iraqia Shipping	3.906671587	4	4.22	70.12
51	COSCO Shpg Energy	3.906671587	13	4.05	65.29
52	Elandra Tankers	3.906671587	5	4.64	107
53	Hunter Tankers	3.906671587	8	4.26	84.34
54	Hyundai Glovis	3.906671587	1	5.25	190
55	Meiji Shipping	3.906671587	3	4.23	79.83
56	Okeanis Eco Tankers	3.906671587	15	4.32	87.2
57	Pentacontinent	3.906671587	1	3.58	35
58	Pertamina	3.906671587	2	4.09	59.25
59	SFL Corporation	3.906671587	5	4.33	81.6
60	Srithai Marine	3.906671587	1	4.13	61
61	Tanker Ventures Ltd	3.906671587	1	3.43	30
62	U-Ming Marine	3.906671587	2	4.58	112.5
63	Yinson Production	3.906671587	5	4.07	63.3
64	Bahri	3.89759426	9	4.06	65.39
65	Brightoil Petroleum	3.893118559	1	4.62	100
66	Almi Tankers	3.892975691	2	4.18	70
67	CSET Tanker Dalian	3.892432668	4	3.88	47.75
68	HMM	3.892326975	9	4.25	82
69	Frontline	3.888961286	31	4.24	79.15
70	Oriental Shipping	3.886526994	1	4.98	145
71	Sonatrach Petroleum	3.883572196	3	4.42	83.67
72	Nippon Yusen Kaisha	3.882451019	6	4.34	80.42
73	Navios Acquisition	3.879654588	28	4	63.21
74	Altomare SA	3.87794125	15	4.02	57.33
75	Oak Maritime Canada	3.86325658	4	4.35	82.12
76	Eurotankers	3.853383052	10	4.04	60.1
77	CSET Tanker Shanghai	3.846604412	1	3.81	44
78	Athenian Carriers	3.844253622	27	4.31	82.39
79	Chandris (Hellas)	3.840283139	6	4.06	70.12
80	Cido Shipping	3.830967831	13	4.24	82.33
81	Nan Fung Shipping	3.830144809	2	4.01	54.25
82	TRF	3.815082005	6	4.15	73.92
83	Wah Kwong Shpg	3.810298515	4	4.02	63.38
84	Tsakos Energy Nav	3.808020694	3	4.15	73.5
85	Kyoei Tanker	3.803127118	3	4.33	85.67
86	Samos Steamship	3.788049031	2	3.7	39.25
87	Neda Maritime Agency	3.759839922	30	4.09	67.91
88	Foresight	3.699037448	1	3.71	40
89	GC Tankers	3.694700061	1	3.86	46.5
90	Capital Ship Mgmt	3.680376848	27	4.26	77.45
91	NS United KK	3.632015071	1	4.01	54
92	Idemitsu Tanker	3.583518938	2	3.78	43.75

Note: Top 10 most active owners are highlighted.

Table A2.6 - Owner encoded values vs Original categorical values (Test set)

A3. Residual Test from GAM model

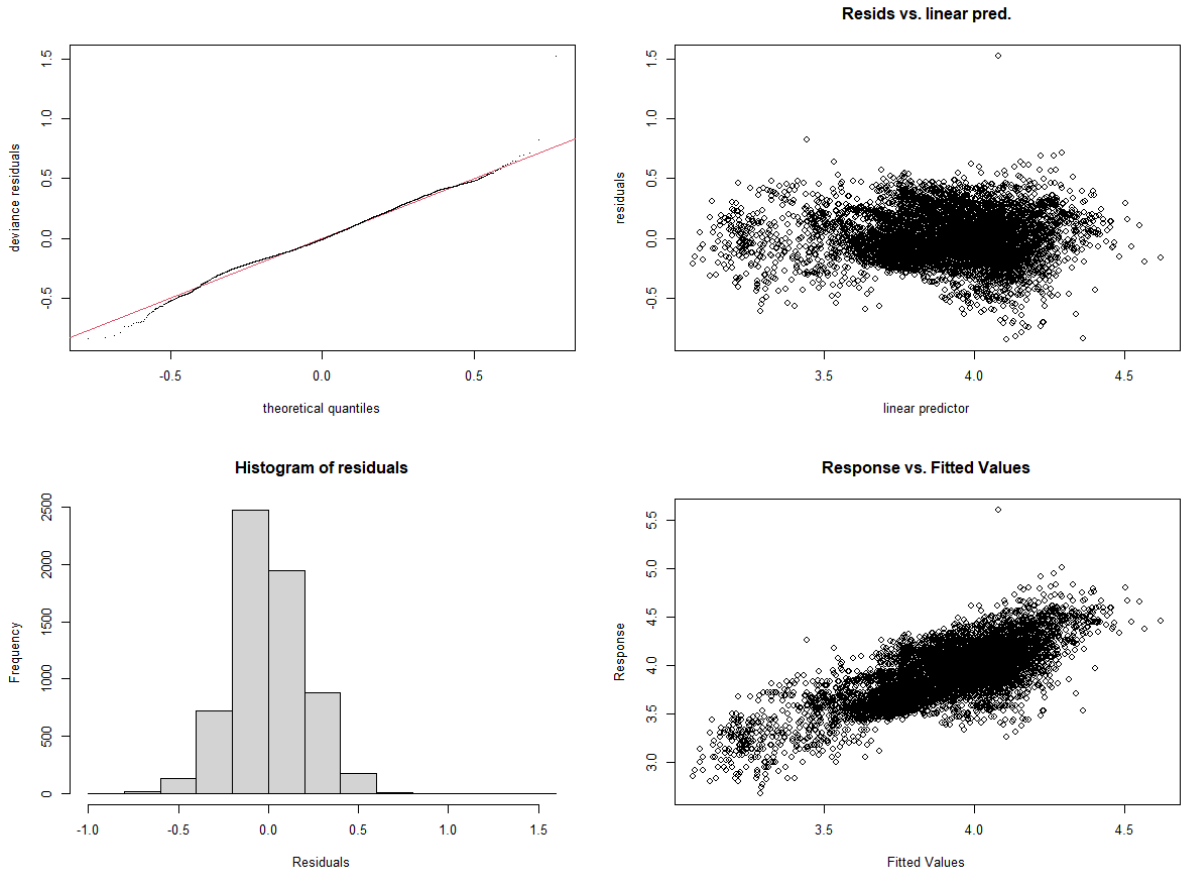


Figure A3.1 - Overview of residuals from GAM model

A4. XGBoost Feature Importance Score

Variables	Gain	Cover	Frequency
Market index	0.556	0.219	0.231
Route	0.272	0.072	0.070
Utilization ratio	0.052	0.279	0.252
Charterer identity	0.036	0.113	0.108
Lead time	0.035	0.105	0.129
Owner identity	0.032	0.137	0.131
Vessel age	0.017	0.074	0.079

Note: XGBoost feature importance scores are based on gain scores in which gain scores of all features sum up to 1 (or 100%).

Table A4.1 - XGBoost Feature Importance Score

A5. XGBoost Interaction

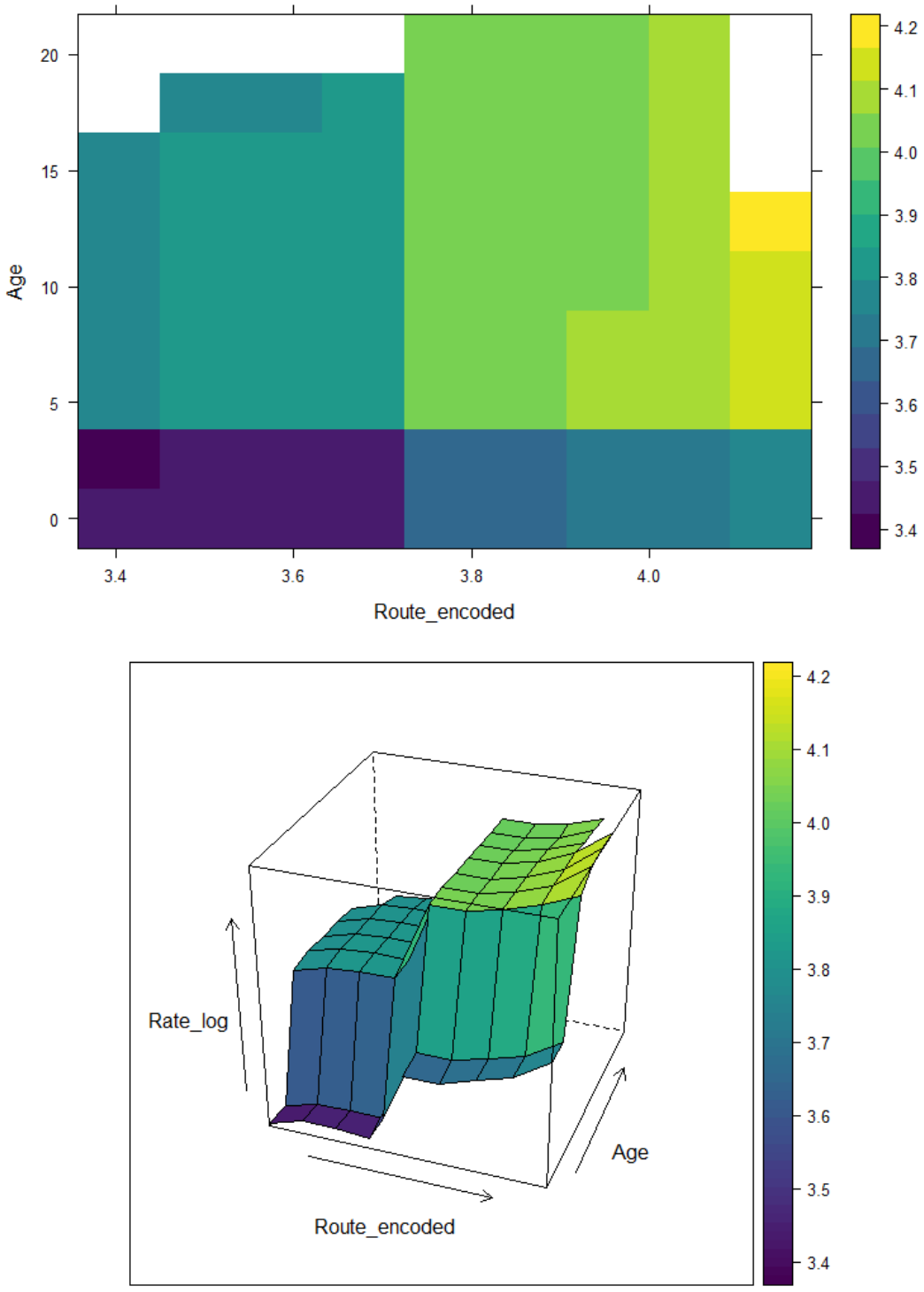


Figure A5.1 - Interaction plot between route and age

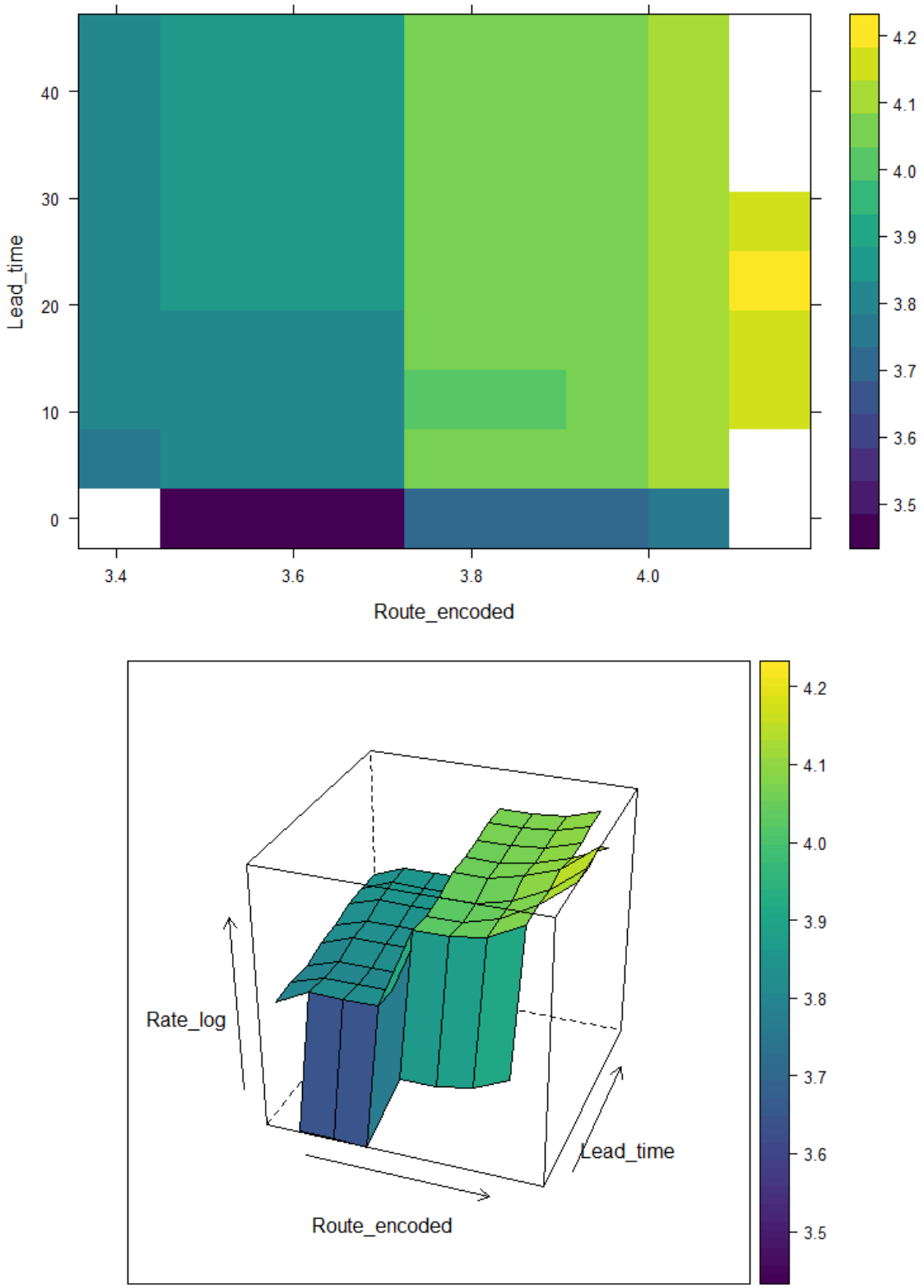


Figure A5.2 - Interaction plot between route and lead time

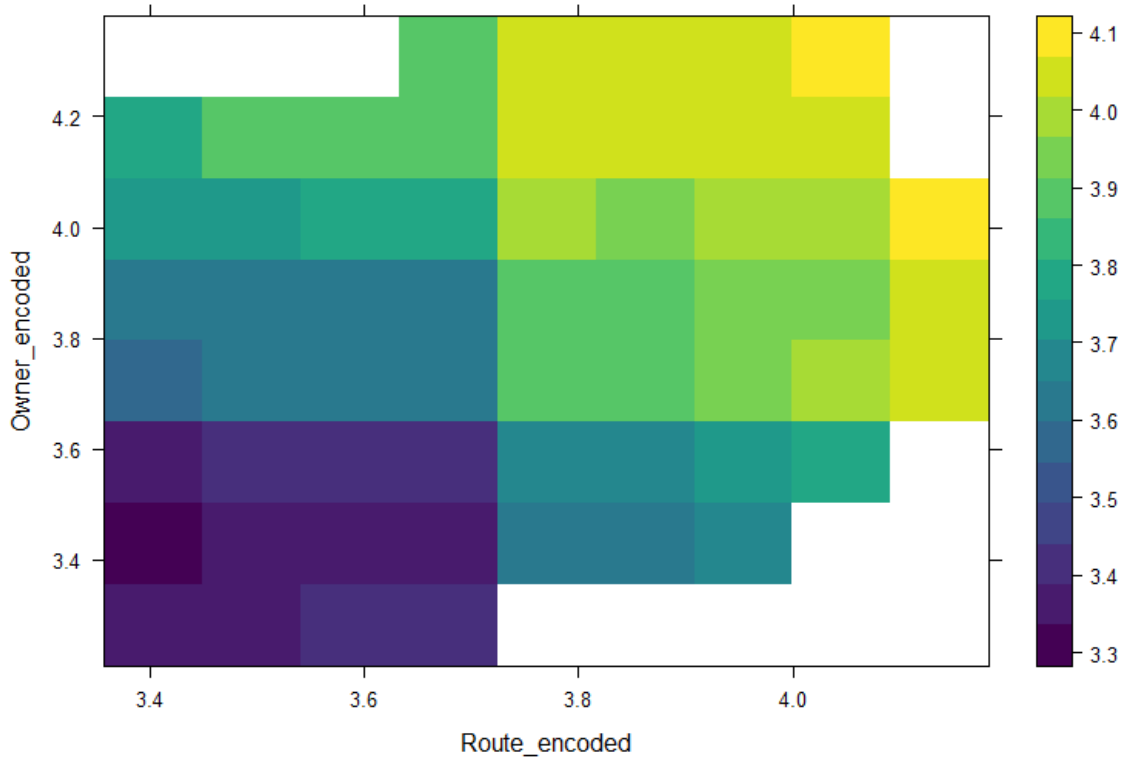


Figure A5.3 - Interaction plot between route and owner

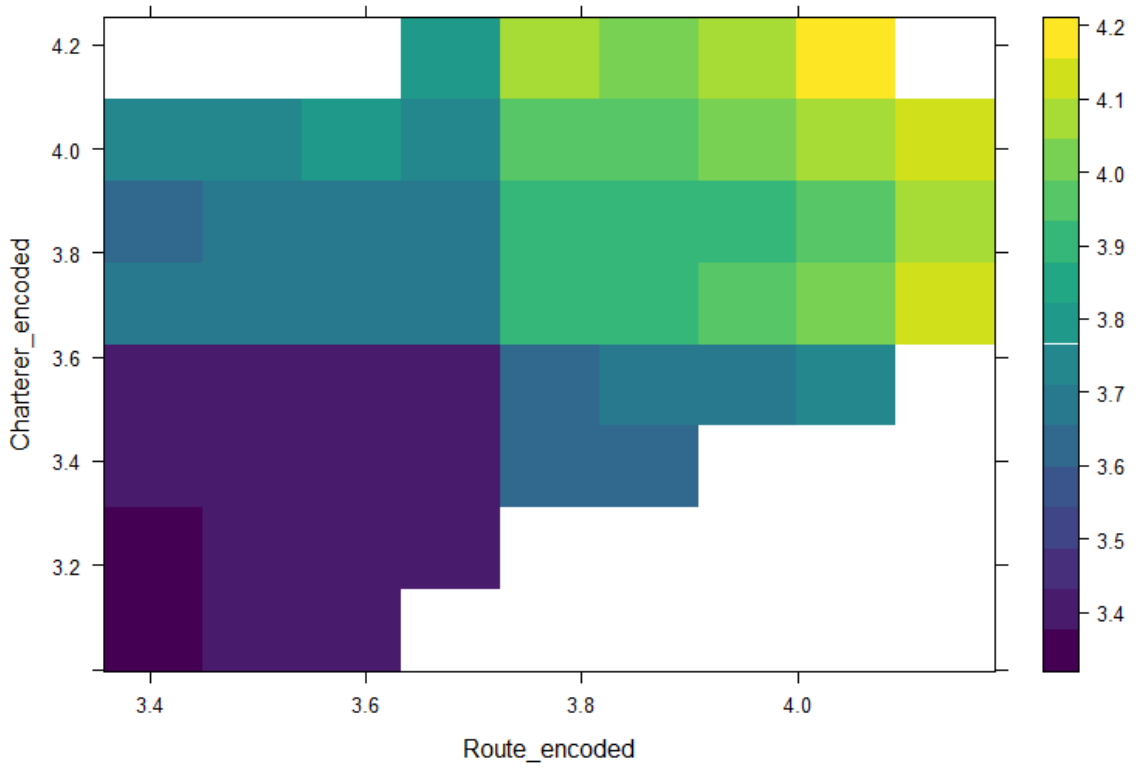


Figure A5.4 - Interaction plot between route and owner

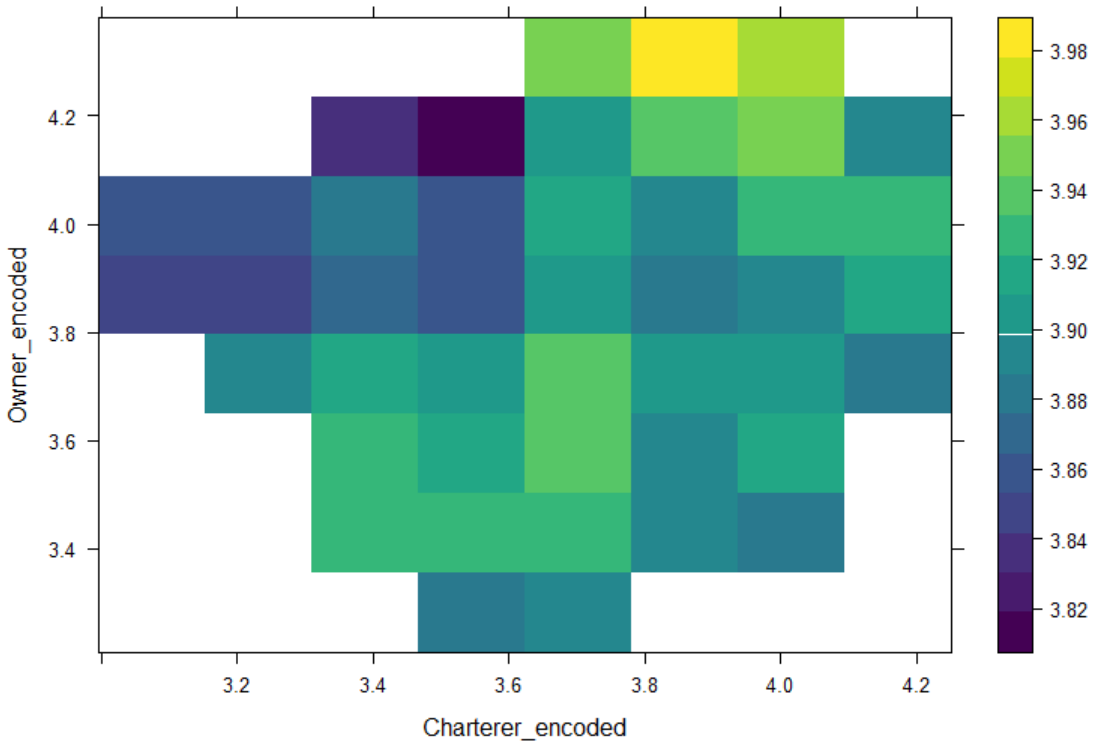


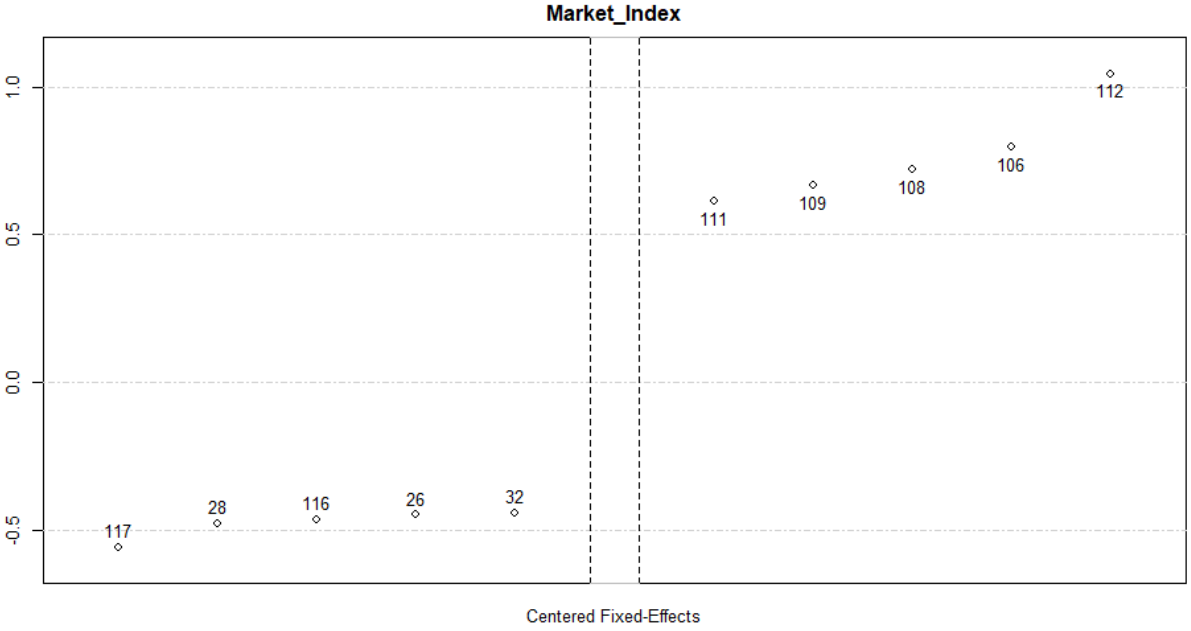
Figure A5.5 - Interaction plot between charterer and owner

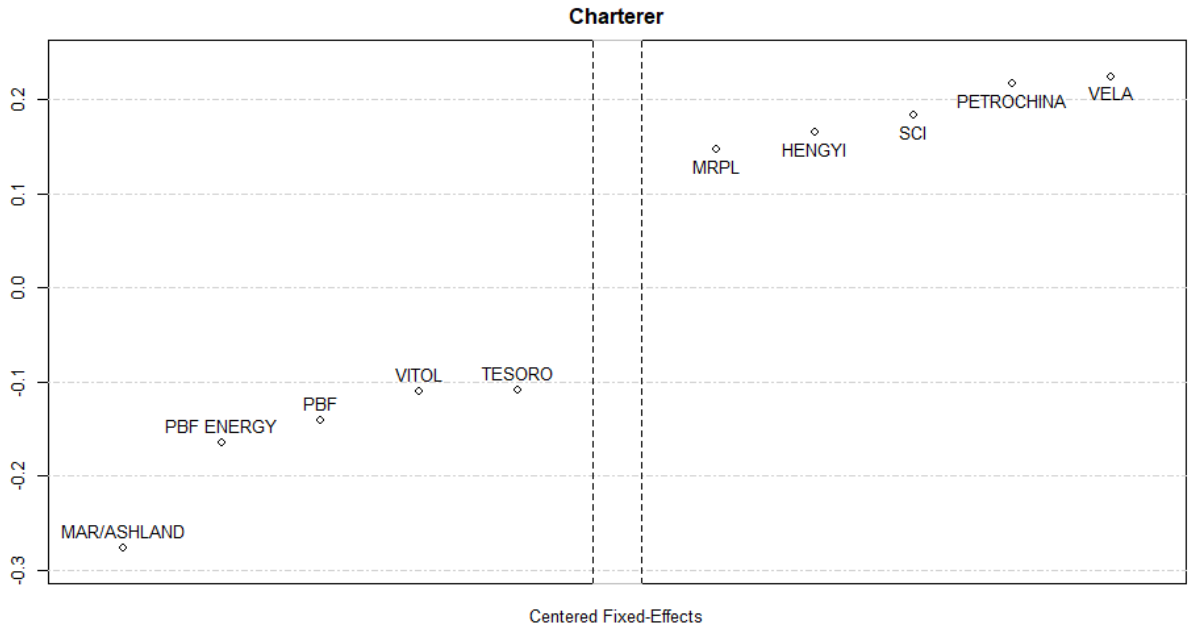
A6. Fixed Effect Linear Regression Model

Variables	Model 1		Model 2		Model 3	
	Coef	t-value	Coef	t-value	Coef	t-value
Lead time (days)	0.0053	1.4105	0.0256	1.5021	0.0061	1.4557
Vessel age (years)	0.03287***	6.9083	0.0265	1.3492	0.03414***	6.3704
Vessel age square (years)	-0.04621***	-7.5227	-0.04344*	-2.1272	-0.04618***	-6.9317
Utilization ratio (%)	-0.0005	-0.1821	0.0012	0.2591	0.0001	0.0190
PERSIAN GULF-CANADA	-0.5405***	-60.8280	-0.7991***	-7.0061	-0.5915***	-22.2550
PERSIAN GULF-CHINA	-0.06165***	-9.9735	0.0522	1.6742	-0.05468***	-5.4044
PERSIAN GULF-EAST	-0.07019***	-9.8848	0.0071	0.2150	-0.06657***	-5.0692
PERSIAN GULF-INDIA	0.08203***	9.9572	0.1042	1.3824	0.03635.	1.9677
PERSIAN GULF-JAPAN	-0.06292***	-7.7869	0.0084	0.2727	-0.04522***	-3.8790
PERSIAN GULF-MALAYSIA	-0.04505***	-5.0938	0.0141	0.3319	-0.0539*	-2.5004
PERSIAN GULF-SINGAPORE	-0.06424***	-8.5211	-0.0602	-1.6902	-0.07151***	-5.1300
PERSIAN GULF-SOUTH AFRICA	-0.1037***	-14.8600	-0.08313*	-2.4298	-0.1197***	-9.8444
PERSIAN GULF-SOUTH KOREA	-0.1303***	-18.4920	-0.0620	-1.4989	-0.09514***	-7.2714
PERSIAN GULF-TAIWAN	-0.1011***	-15.5500	-0.0066	-0.2246	-0.05249***	-4.1761
PERSIAN GULF-THAI LAND	-0.05903***	-6.8141	-0.0751*	-2.4654	-0.0355	-1.3346
PERSIAN GULF-UKC	-0.5588***	-65.3070	-0.5072***	-20.3800	-0.5627***	-37.6030
PERSIAN GULF-USG	-0.6146***	-73.4990	-0.5915***	-20.8340	-0.6267***	-39.6480
PERSIAN GULF-WEST	-0.57***	-52.0980	-0.6405***	-20.4380	-0.5839***	-29.6740
WEST AFRICA-CHINA	-0.04085***	-11.1660	-0.0047	-0.1890	-0.03755**	-3.6679
WEST AFRICA-EAST	-0.03622***	-5.7484	-0.0213	-0.8541	-0.03173*	-2.5568
WEST AFRICA-INDIA	0.02083.	1.9249	0.0172	0.5385	-0.0166	-1.1433
WEST AFRICA-TAIWAN	-0.05404***	-6.4334	0.0031	0.1243	-0.03024*	-2.1837
WEST AFRICA-UKC	0.1167***	26.4940	0.1459***	7.3198	0.1178***	12.8270
WEST AFRICA-USG	0.0172	0.9873	0.0221	0.6297	0.0067	0.3180
Fixed Effects:						
Market index (monthly unit)	YES		NO		YES	
Charterer	NO		YES		YES	
Owner	NO		YES		YES	
S.E type: Clustered	by ROUTE		by ROUTE		by ROUTE	
Number of observations	7485		7485		7485	
R^2	0.8564		0.3376		0.8691	

Note: Signif. codes are respectively: 0'***' 0.001'**' 0.01'*' 0.05'.' 0.1" 1.

Table A6.1 - Estimation of the log freight rate based on fixed effect regression.


Market_Index
1.0
0.5
0.0
-0.5
112
106
108
109
111
28
116
26
32
117
Centered Fixed-Effects


Charterer
0.2
0.1
0.0
-0.1
-0.2
-0.3
PETROCHINA
VELA
SCI
HENGYI
MRPL
VITOL
TESORO
PBF
PBF ENERGY
MAR/ASHLAND
Centered Fixed-Effects

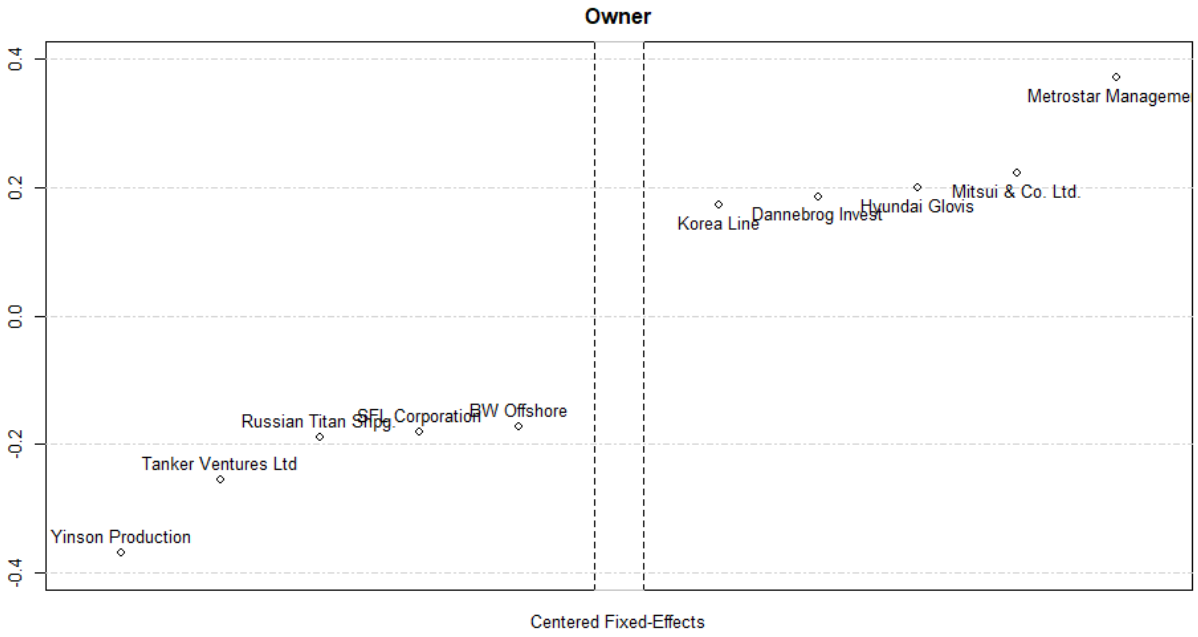


Figure A6.1 - The most notable fixed-effects in the model (3) - Time fixed effect (monthly level) and charterer and owner fixed effect

A7. Results from GAM Model for full dataset

Parametric coefficients		
	Estimate	Significance
Intercept	3.945	***
Smooth terms		
	Effective DF	Significance
s(Lead time)	7.107	***
s(Vessel age)	2.465	***
s(Utilization ratio)	4.522	***
s(Market index)	7.955	***
s(Charterer identity)	4.877	***
s(Owner identity)	3.947	***
s(Route)	5.817	***
REML score	682.46	
R^2	0.455	
N	7485	

Note: Signif. codes are respectively: 0'***' 0.001'**' 0.01'*' 0.05'.' 0.1" 1.

Table A7.1 - Smooth terms from GAM model

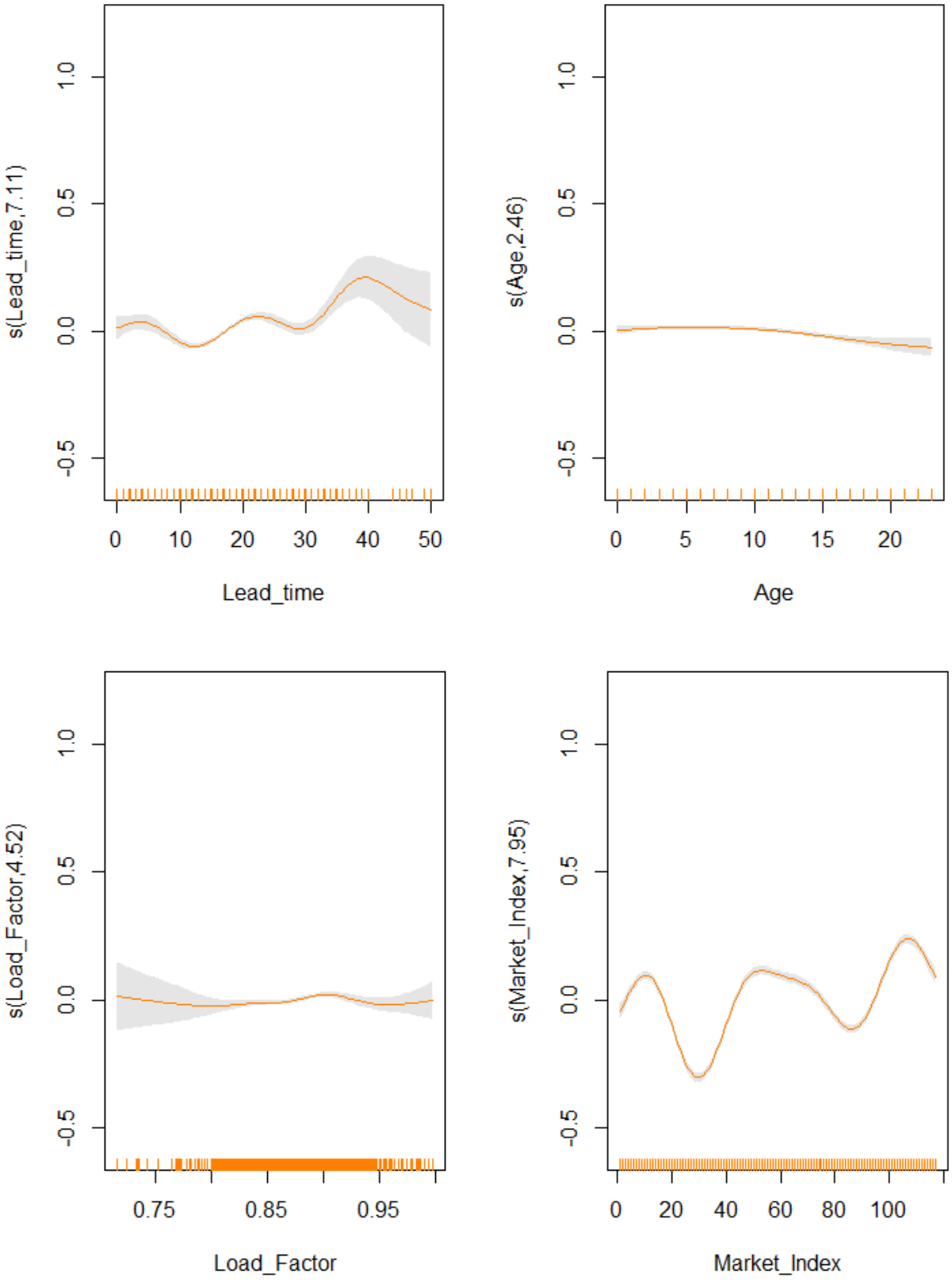


Figure A7.1 - Smooth of GAM model

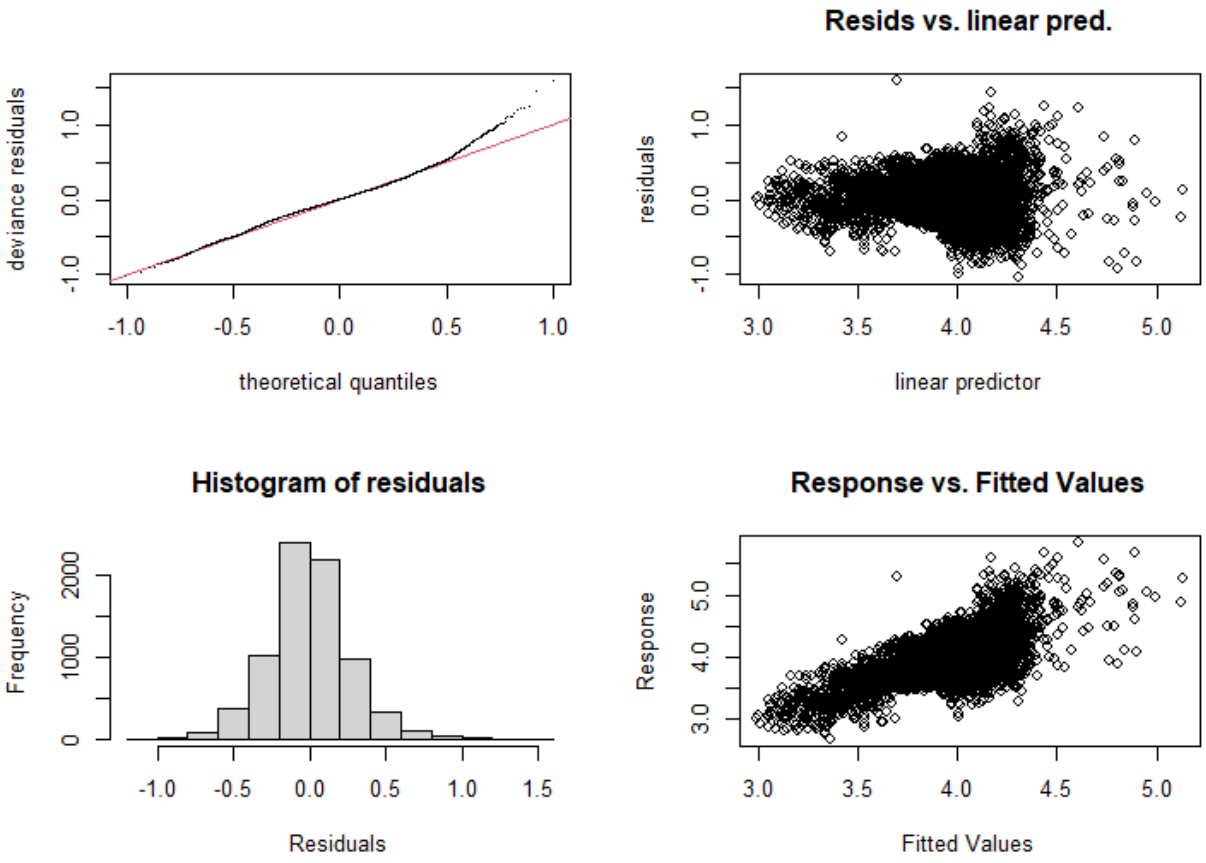
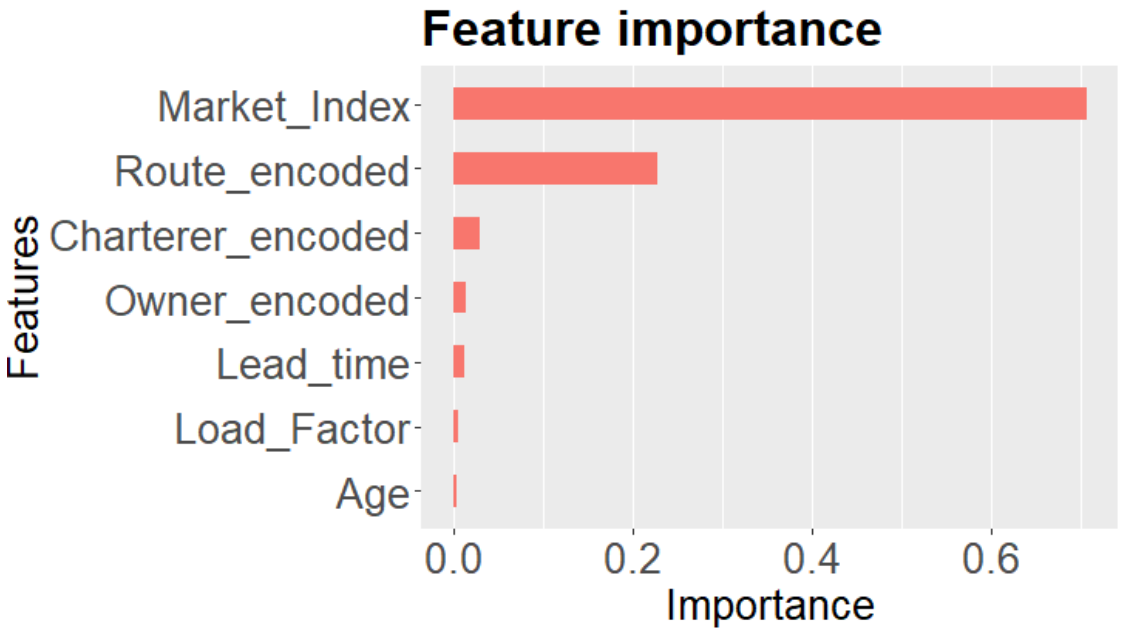


Figure A7.2 - Overview of residuals from GAM model

A8. Results from XGBoost Model for full dataset



Note: XGBoost feature importance scores are based on gain scores in which gain scores of all features sums up to 1 (or 100%).

Figure A8.1 - Feature importance based on XGBoost framework

Variables	Gain	Cover	Frequency
Market index	0.706	0.648	0.663
Route	0.228	0.103	0.079
Charterer identity	0.030	0.056	0.053
Owner identity	0.015	0.063	0.061
Lead time	0.012	0.055	0.058
Utilization ratio	0.006	0.047	0.053
Vessel age	0.004	0.028	0.034

Note: XGBoost feature importance scores are based on gain scores in which gain scores of all features sums up to 1 (or 100%).

Table A8.1 - XGBoost Feature Importance Score

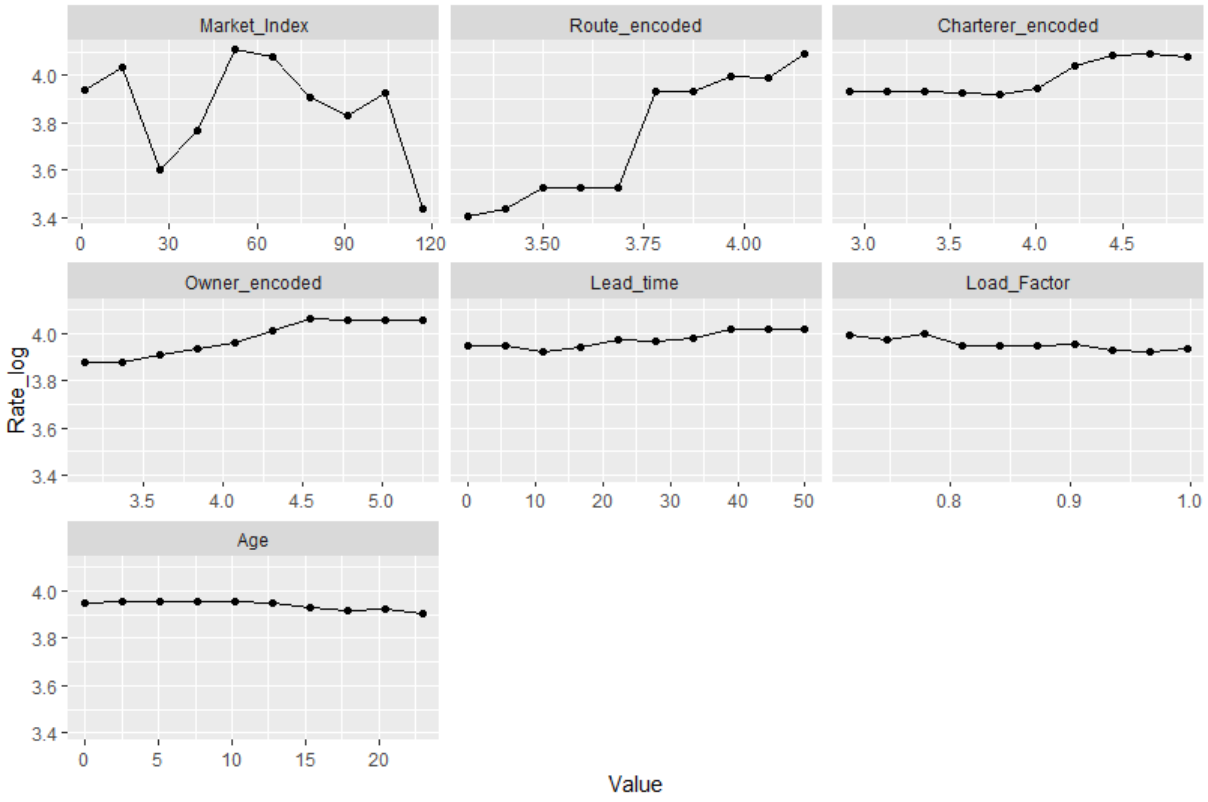


Figure A8.2 - Partial dependence plots from XGBoost model

A9. R Code

```
##################################
# Master Thesis
##################################
getwd()
setwd("C:/Users/47462/OneDrive/Documents/NHH/Master Thesis/VLCC")
# Import necessary packages
library(readxl)
library(dplyr)
library(ggplot2)
library(tidyr)
library(hrbrthemes)
library(viridis)
library(babynames)
library(tidyverse)
library(plotly)
library(zoo)
library(anytime)
library(forecast)
library(data.table)
library(xgboost)
library(SHAPforxgboost) #SHAP Value
library(mlr) #Tuning hyperparameters
library(Metrics)
```

```
library(Hmisc)
library(checkmate)
library(e1071)
library(data.table)
library(forcats)
library(lubridate)
library(cowplot)
library(corrplot) #Check multicollinearity
library(plotmo)
library(ExPanDaR) #Data descriptive analysis
library(gridExtra)
library(fixest) #Fixed effect
library(caret)
library(timetk) #time serie cross validation
library(mgcv) #GAM
#####################################################
# 3. Data
# 3.1. Data Collection
#####################################################
# Import data files
VLCC_2011 <- read_excel("Clarkson VLCC fixtures.xlsx",5)
VLCC_2012 <- read_excel("Clarkson VLCC fixtures.xlsx",4)
VLCC_2013 <- read_excel("Clarkson VLCC fixtures.xlsx",3)
VLCC_2014 <- read_excel("Clarkson VLCC fixtures.xlsx",2)
# Dataset 2015 has a different format then other dataset. Therefore, we need to adjust format and change column names
VLCC_2015 <- read_excel("Clarkson VLCC fixtures.xlsx",1)
```

```
colnames(VLCC_2015)[colnames(VLCC_2015) == 'Laycan_From'] <- "Laycan From"

colnames(VLCC_2015)[colnames(VLCC_2015) == 'Laycan_To'] <- "Laycan To"

VLCC_2015$Date <- anydate(VLCC_2015$Date)

VLCC_2015$Dwt <- as.numeric(as.character(VLCC_2015$Dwt))

VLCC_2016 <- read_excel("vlcc_2016.xlsx")

VLCC_2017 <- read_excel("vlcc_2017.xlsx")

VLCC_2018 <- read_excel("vlcc_2018.xlsx")

VLCC_2019 <- read_excel("vlcc_2019.xlsx")

VLCC_2020 <- read_excel("vlcc_2020.xlsx")

# Consolidated data file

VLCC <- Reduce(function(x, y) merge(x, y, all=TRUE),

           list(VLCC_2011, VLCC_2012, VLCC_2013, VLCC_2014, VLCC_2015, VLCC_2016, VLCC_2017, VLCC_2018, VLCC_2019, VLCC_2020))

df <- VLCC[,-c(8,22:26)]

# Take a look at dataset

head(df)



#############################################################################

# 3.2. Data Pre-processing

#############################################################################

# Convert column Dwt to numeric class

df$Dwt <- as.numeric(as.character(df$Dwt))

# Add column Lead_time which is the difference between Laycan From and transaction Date

df$Lead_time<- as.numeric(as.character(difftime(df$`Laycan From` ,df$`Date` , units = c("days"))))

# Combine all Route which appeared in less than 20 transactions into "Other"

df$Route[df$Route %in%  names(table(df$Route))[table(df$Route) <20]] = "OTHER"

df <- df %>%
```

```
  # Omit values without charterer's names, owner's names, quantity, DWT, ages
  na.omit(cols = c(Charterer, Owner, Quantity, Built, Dwt)) %>%
  # Calculate Age, Year and Load Factor
  mutate(Age = as.numeric(format(as.Date(Date), format = "%Y")) - Built,
         Load_Factor = Quantity/Dwt,
         Year = format(as.Date(Date), format = "%Y")) %>%
  # Only chose fixture with rate expressed in WS
  filter(Unit == "WS")
df <- df[df$Load_Factor <= 1,] # Remove rows with Load_Factor more than 1
df <- df[df$Load_Factor >= 0.7,] # Remove rows with Load_Factor less than 0.7
df <- df[df$Lead_time >= 0,] # Remove rows with negative Lead_time
df <- df[df$Lead_time <=50,] # Remove rows with Lead_time more than 50 days
# Plot the freight rate
ggplot(df) + geom_boxplot(aes(x=Year, y=Rate))+
  theme(text = element_text(size=20))
# Check the distribution of the response variable
blog <- ggplot(data = df) +
  geom_histogram(aes(x = Rate, y=..density..), fill="steelblue", colour="black") +
  ggtitle("Distribution of Rate before transformation") +
  stat_function(fun = dnorm,
                args = list(mean = mean(df$Rate), sd = sd(df$Rate)),
                color = "black", size = 1)
# Log transformation of the target variable
df <- df %>%
  mutate (Rate_log = log(Rate +1))
# Check the distribution of the response variable after transformation
```

```
alog <- ggplot(data = df) +
  geom_histogram(aes(x = Rate_log, y=..density..), fill="steelblue", colour="black") +
  ggtitle("Distribution of Rate after log_transformation") +
  stat_function(fun = dnorm,
                args = list(mean = mean(df$Rate_log), sd = sd(df$Rate_log)),
                color = "black", size = 1)
# Combine three plots into same page
plot_grid(blog, alog, labels = "AUTO")
# Add column "Market index"
diffMarket_Index <- function(end_date, start_date) {
  end <- as.POSIXlt(end_date)
  start <- as.POSIXlt(start_date)
  12 * (end$year - start$year) + (end$mon - start$mon)
}
for (i in 1:nrow(df)) {
  df$Market_Index[i] <- diffMarket_Index(df$Date[i],df$Date[1])+1
}


#####################################################################
# 3.3. Descriptive Statistics
#####################################################################
# Data description
ExPanD(df)
# Check multicollinearity of numeric variables
cor_numVar <- cor(df[,c(21:23,25:26)], use="pairwise.complete.obs")
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
```

```
#########
# Age, Lead_time and Load_Factor
# Histogram
df[,c(21:23,26)] %>%
  gather() %>%
  ggplot(aes(value,fill=key)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram() +
  theme(legend.position="none")+ theme(axis.title.x = element_text(size = 20),
                                       axis.title.y = element_text(size = 20),
                                       axis.text.x = element_text(size=20),
                                       axis.text.y = element_text(size=20))
# Plot scatterplot between Rate and other numeric variables
plot_lf <- ggplot(df, aes(x = Load_Factor, y = Rate)) +
  geom_point(color="red") +
scale_color_viridis_d() + theme(axis.title.x = element_text(size = 20),
                                axis.title.y = element_text(size = 20),
                                axis.text.x = element_text(size=20),
                                axis.text.y = element_text(size=20))
plot_age <- ggplot(df, aes(x = Age, y = Rate)) +
  geom_point(color="red") +
  scale_color_viridis_d()+ theme(axis.title.x = element_text(size = 20),
                                 axis.title.y = element_text(size = 20),
                                 axis.text.x = element_text(size=20),
                                 axis.text.y = element_text(size=20))
plot_lt <- ggplot(df, aes(x = Lead_time, y = Rate)) +
```

```
  geom_point(color="red") +
  scale_color_viridis_d()+ theme(axis.title.x = element_text(size = 20),
                                 axis.title.y = element_text(size = 20),
                                 axis.text.x = element_text(size=20),
                                 axis.text.y = element_text(size=20))
# Combine three plots into same page
plot_grid(plot_lf, plot_age, plot_lt, labels = "AUTO")
############
# Rate over time
df_mean_Month <- df %>%
  group_by(Market_Index) %>%
  summarise(Mean_Rate = round(mean(Rate),2))
ggplot(df_mean_Month, aes(x = Market_Index, y = Mean_Rate)) + xlab("Month") +
  geom_line() +
  theme_bw()
############
# Charterers
# Top ten Charterers with highest numbers of transactions
ch <- df %>%
  group_by(Charterer) %>%
  summarise(Fixtures = n(),
            Rate = round(mean(Rate),2)) %>%
  arrange(desc(Fixtures)) %>% # Arrange number of fixtures
  top_n(n=10, Fixtures) # Choose top ten fixtures
# Number of transactions which other charterers participated in
m <- df %>%
```

```
  filter(!Charterer %in% ch$Charterer) %>%
  summarise(Fixtures = n(),
            Rate = round(mean(Rate),2))
# Add column "Others" and "Total" which account for total transactions made by other charterers and overall charterers
ch[11,] = c("Others", m[1,])
ch[12,] = list("Total", nrow(df),
               round(mean(df$Rate),2))
# Add column "Percent" and "Cumul." which are percentage of fixtures and cumulative percentage of fixtures, respectively
ch <- ch%>%
  mutate(Percent = round((Fixtures/nrow(df)*100),1),
         Cumul. = NA) # Assign column "Cumul." as NA initially
ch <- as.data.frame(ch) # Convert ch to data frame
# Calculate "Cumul."
# Assign the first row of the column equals to the percentage of that transaction
ch[1,5] <- ch[1,4]
# From the second rows going forward, the value of cumulative percentage will equal to the previous cumulative percentage
# plus the percentage of this transaction
for (i in c(2:(nrow(ch)-1))){
  ch[i,5] <- ch[i-1,5] + ch[i,4]}
# Owners
# We do the same process for Top ten Owners:
o <- df %>%
  group_by(Owner) %>%
  summarise(Fixtures = n(),
            Rate = round(mean(Rate),2)) %>%
```

```
  arrange(desc(Fixtures)) %>%
  top_n(n=10, Fixtures)
m <- df %>%
  filter(!Owner %in% o$Owner) %>%
  summarise(Fixtures = n(),
            Rate = round(mean(Rate),2))
o[11,] = c("Others", m[1,])
o[12,] = list("Total", nrow(df),
              round(mean(df$Rate),2))
o <- o%>%
  mutate(Percent = round((Fixtures/nrow(df)*100),1),
         Cumul. = NA)
o <- as.data.frame(o)
o[1,5] <- o[1,4]
for (i in c(2:(nrow(o)-1))){
  o[i,5] <- o[i-1,5] + o[i,4]}
# Dataframe with top ten owner and charterer:
df_pair <- df %>%
  filter(Charterer %in% ch$Charterer) %>%
  filter(Owner %in% o$Owner) %>%
  group_by(Owner, Charterer) %>%
  summarise(Fixtures = n())
# Create heatmap with top ten owner and charterer:
ggplot(df_pair, aes(Owner, Charterer)) +
  geom_tile(aes(fill = Fixtures)) + geom_text(aes(label = Fixtures), size = 8) +
  theme(axis.title.x = element_text(size = 20),
```

```
      axis.title.y = element_text(size = 20),
      axis.text.x = element_text(angle=45, hjust=1, size=22),
      axis.text.y = element_text(size=22),
      legend.text = element_text(size=20),
      legend.title = element_text(size=20),
      aspect.ratio = 2/3)
##############
# Top 10 routes
df_route <- df %>%
  group_by(Route) %>%
  summarise(Fixtures = n(), # number of transaction of that each route
            # Summarise mean of relative features corresponding to each route, round the result
            Rate = round(mean(Rate),2),
            UR = round(mean(Load_Factor),2),
            Age = round(mean(Age),2),
            Leadtime = round(mean(Lead_time),2)) %>%
  arrange(desc(Fixtures)) %>% # Arrange number of fixtures
  top_n(n=10, Fixtures) # Choose top ten fixture
# Summary of top 10 routes and all routes by number of fixtures of each route over time
df_route_plot <- df %>%
  mutate(Year = year(Date)) %>%
  filter(Route %in% df_route$Route) %>%
  group_by(Year, Route) %>%
  summarise(Fixtures = n())
# Summarise total  of fixture, mean of other features of other routes
n <- df %>%
```

```
  filter(!Route %in% df_route$Route) %>%
  summarise(Fixtures = n(),
            Rate = round(mean(Rate),2),
            UR = round(mean(Load_Factor),2),
            Age = round(mean(Age),2),
            Leadtime = round(mean(Lead_time),2))
# Add column "Others" and "All" and calculate corresponding values of all columns
df_route[11,] <- c("Others", n[1,])
df_route[12,] <- list("All", nrow(df),
                      round(mean(df$Rate),2),
                      round(mean(df$Load_Factor),2),
                      round(mean(df$Age),2),
                      round(mean(df$Lead_time),2))
#plots the stacked area chart of top ten route
p <- df_route_plot %>%
  ggplot(aes(x=Year, y=Fixtures, fill=Route, text=Route))+
  geom_area()+
  scale_fill_viridis(discrete = T)+
  theme(legend.position = 'none') +
  theme_ipsum()+
  theme(axis.title.x = element_text(size = rel(1.8)),
        axis.title.y = element_text(size = rel(1.8)),
        text = element_text(size=12))
ggplotly(p, tootltip='text')
# Plot ranking of routes based on rates
df_rr <- df %>%
```

```
  group_by(Route) %>%
  summarise(mean_Rate = mean(Rate)) %>%
  mutate(variable = NA)
df_rr <- as.data.frame(df_rr)
# Plot
ggplot(df_rr, aes(mean_Rate, reorder(Route, -mean_Rate))) +
  geom_bar(stat="identity", fill = "red") + ylab(label = "Route") +
  theme(axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15),
        axis.text.x = element_text(size=15),
        axis.text.y = element_text(size=15),
        legend.text = element_text(size=15),
        legend.title = element_text(size=15))


#####################################################################
# 5. Methodology
# 5.1. Preparation before modeling
#####################################################################
# 5.1.1. Split data into training and test data
# Training data consists of observations from 2011 to 2018.
# Test set covers the last two year period
train_index <- 2011:2018
test_index <- 2019:2020
train_temp <- df[which(df$Year %in% train_index),]
test_temp <- df[which(df$Year %in% test_index),]
```

```
########################################

# 5.1.2. Target Encoding

# Function to encode data

encode_target <- function(x, y, sigma = NULL) {

  t1 <- aggregate(y, list(factor(x, exclude = NULL)), mean, na.rm = TRUE)

  t2 <- t1[is.na(as.character(t1[, 1])), 2]

  t3 <- t1[, 2]

  names(t3) <- t1[, 1]

  t3 <- t3[x]

  t3[is.na(t3)] <- t2

  if (!is.null(sigma)) {

    t3 <- t3 * rnorm(length(t3), mean = 1, sd = sigma)

  }

  t3

}

train_temp[["Route_encoded"]] <- encode_target(train_temp[["Route"]], train_temp[["Rate_log"]])

train_temp[["Charterer_encoded"]] <- encode_target(train_temp[["Charterer"]], train_temp[["Rate_log"]])

train_temp[["Owner_encoded"]] <- encode_target(train_temp[["Owner"]], train_temp[["Rate_log"]])

# Match encoded data of train set to test set

test_temp[["Route_encoded"]] <- train_temp$Route_encoded[match(test_temp$Route,train_temp$Route)]

test_temp[["Charterer_encoded"]] <- train_temp$Charterer_encoded[match(test_temp$Charterer,train_temp$Charterer)]

test_temp[["Owner_encoded"]] <- train_temp$Owner_encoded[match(test_temp$Owner,train_temp$Owner)]

# Fill in NA of test sets by global mean of train set

test_temp$Charterer_encoded[is.na(test_temp$Charterer_encoded)] <- mean(train_temp$Rate_log)
```

```
test_temp$Owner_encoded[is.na(test_temp$Owner_encoded)] <- mean(train_temp$Rate_log)

test_temp$Route_encoded[is.na(test_temp$Route_encoded)] <- mean(train_temp$Rate_log)

# Get the final data

train <- train_temp %>%

  select(Date, Charterer_encoded, Route_encoded, Owner_encoded, Lead_time, Age, Load_Factor, Market_Index, Rate_log)

test <- test_temp %>%

  select(Charterer_encoded, Route_encoded, Owner_encoded, Lead_time, Age, Load_Factor, Market_Index, Rate_log)

# Table of original data vs encoded data

# Routes

train_route_encoded <- train_temp %>%

  group_by(Route) %>%

  summarise(Route_encoded = mean(Route_encoded),

            Fixtures = n(),

            Average_Rate_log = round(mean(Rate_log),2),

            Average_Rate = round(mean(Rate),2)) %>%

  arrange(desc(Route_encoded))

test_route_encoded <- test_temp %>%

  group_by(Route) %>%

  summarise(Route_encoded = mean(Route_encoded),

            Fixtures = n(),

            Average_Rate_log = round(mean(Rate_log),2),

            Average_Rate = round(mean(Rate),2)) %>%

  arrange(desc(Route_encoded))

# Charterer

train_charterer_encoded <- train_temp %>%
```

```
  group_by(Charterer) %>%
  summarise(Charterer_encoded = mean(Charterer_encoded),
            Fixtures = n(),
            Average_Rate_log = round(mean(Rate_log),2),
            Average_Rate = round(mean(Rate),2)) %>%
  arrange(desc(Charterer_encoded))
test_charterer_encoded <- test_temp %>%
  group_by(Charterer) %>%
  summarise(Charterer_encoded = mean(Charterer_encoded),
            Fixtures = n(),
            Average_Rate_log = round(mean(Rate_log),2),
            Average_Rate = round(mean(Rate),2)) %>%
  arrange(desc(Charterer_encoded))
# Owner
train_owner_encoded <- train_temp %>%
  group_by(Owner) %>%
  summarise(Owner_encoded = mean(Owner_encoded),
            Fixtures = n(),
            Average_Rate_log = round(mean(Rate_log),2),
            Average_Rate = round(mean(Rate),2)) %>%
  arrange(desc(Owner_encoded))
test_owner_encoded <- test_temp %>%
  group_by(Owner) %>%
  summarise(Owner_encoded = mean(Owner_encoded),
            Fixtures = n(),
            Average_Rate_log = round(mean(Rate_log),2),
```

```
          Average_Rate = round(mean(Rate),2)) %>%
  arrange(desc(Owner_encoded))


################################################
# 5.2. Fitting models
# Time series cross-validation
tscv <- train %>%
  time_series_cv(
    date_var    = Date,
    initial     = "3 years",
    assess      = "1 year",
    skip        = "1 year",
    slice_limit = 10
  )
# Plot cross-validation plan
tscv %>%
  plot_time_series_cv_plan(Date, Rate_log, .interactive = FALSE, .title = "Cross Validation Plan")
# Divide data into 4 folds:
# Fold 1:
train1 <- train[c(tscv$splits[[1]]$in_id[1]:tail(tscv$splits[[1]]$in_id, n=1)),-1]
vali1 <- train[c(tscv$splits[[1]]$out_id[1]:tail(tscv$splits[[1]]$out_id, n=1)),-1]
fold1 <- rbind(train1, vali1)
# Fold 2:
train2 <- train[c(tscv$splits[[2]]$in_id[1]:tail(tscv$splits[[2]]$in_id, n=1)),-1]
vali2 <- train[c(tscv$splits[[2]]$out_id[1]:tail(tscv$splits[[2]]$out_id, n=1)),-1]
fold2 <- rbind(train2, vali2)
```

```
# Fold 3:
train3 <- train[c(tscv$splits[[3]]$in_id[1]:tail(tscv$splits[[3]]$in_id, n=1)),-1]
vali3 <- train[c(tscv$splits[[3]]$out_id[1]:tail(tscv$splits[[3]]$out_id, n=1)),-1]
fold3 <- rbind(train3, vali3)
# Fold 4:
train4 <- train[c(tscv$splits[[4]]$in_id[1]:tail(tscv$splits[[4]]$in_id, n=1)),-1]
vali4 <- train[c(tscv$splits[[4]]$out_id[1]:tail(tscv$splits[[4]]$out_id, n=1)),-1]
fold4 <- rbind(train4, vali4)
#########################################################################
# Prepare for fitting XGBoost model
# The predictor variables
predictors <- train %>%
  select(-c(Rate_log, Date)) %>%
  as.matrix()
# The response variable
output <- train$Rate_log
# The predictor variables from test data
test_output <- test$Rate_log
# Constract xgb.DMatrix object for XGBoost
dtrain <- xgb.DMatrix(data = predictors, label = output)
# Change column names of training, validation and test data
colnames(fold1) <- make.names(colnames(fold1),unique = T)
colnames(fold2) <- make.names(colnames(fold2),unique = T)
colnames(fold3) <- make.names(colnames(fold3),unique = T)
colnames(fold4) <- make.names(colnames(fold4),unique = T)
```

```
#####################################################################

# 6. Results & Discussion

# 6.1.GAM & XGBoost

#####################################################################

# GAM

# Fit model

set.seed(5678)

gam1<-
gam(Rate_log~s(Lead_time)+s(Age)+s(Load_Factor)+s(Market_Index)+s(Charterer_encoded)+s(O
wner_encoded)

        +s(Route_encoded),

        data = train[,-1], method="REML", select = TRUE)

# Summary of GAM results

summary(gam1)

# Plot partial dependence plots

plot(gam1, pages = 4, residuals=F, pch=19, cex=0.25,

    scheme=1, col='#FF8000', shade=T,shade.col='gray90')

# Check GAM residuals

mar <- par(mfrow = c(2, 2))

gam.check(gam1, old.style=FALSE,

        type=c("deviance","pearson","response"))

par(mar)

########################################

# Perform prediction

pred1 = predict(gam1, test)

# Calculate accuracy measures

mse_1 <- mse(pred1, test_output)
```

```
mae_1 <- mae(pred1,  test_output)
rmse_1 <- rmse(pred1,  test_output)
mape_1 <- mape(pred1,  test_output)
cat("MSE: ", mse_1, "MAE: ", mae_1, "RMSE: ", rmse_1, "MAPE: ", mape_1)


#####################################################################
# XGBoost
# Tunning Hypermeters
# create mlr task for XGBoost
trainTask <- makeRegrTask(data = train[,-1], target = "Rate_log")
testTask <- makeRegrTask(data = test, target = "Rate_log")
trainTask1 <- makeRegrTask(data = fold1, target = "Rate_log")
trainTask2 <- makeRegrTask(data = fold2, target = "Rate_log")
trainTask3 <- makeRegrTask(data = fold3, target = "Rate_log")
trainTask4 <- makeRegrTask(data = fold4, target = "Rate_log")
# Create learner object:
learn <- makeLearner(
  "regr.xgboost",  #model type: XGBoost
  predict.type = "response",
  par.vals = list(
    objective = "reg:squarederror",
    eval_metric = "error",
    nrounds = 1000
  )
)
# Impute base learner by median
```

```
learn <- makeImputeWrapper(learn, classes = list(numeric = imputeMedian(), integer = imputeMedian()))

# Define the list parameters for XGBoost model

param <- makeParamSet(

  makeIntegerParam("min_child_weight", lower = 1, upper = 10),

  makeIntegerParam("nrounds", lower = 500, upper = 2000),

  makeIntegerParam("max_depth", lower = 1, upper = 10),

  makeNumericParam("eta", lower = 0.01, upper = 0.3),

  makeNumericParam("subsample", lower = 0.5, upper = 0.8),

  makeNumericParam("lambda", lower = -2, upper = 0, trafo = function(x) 10^x)

)

# Fold 1

set.seed(123)

# Run base learner for the training Task and randomly search through 10 iterations

best_xgb1 <- tuneParams(learn, task = trainTask1,

                        resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train1),

                                                              test.inds = (1+nrow(train1)):nrow(fold1),

                                                              nrow(fold1)),

                        par.set = param,

                        control = makeTuneControlRandom(maxit = 10L))

#########

# Fold 2

set.seed(124)

# Run base learner for the training Task and randomly search through 10 iterations

best_xgb2 <- tuneParams(learn, task = trainTask2,

                        resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train2),

                                                              test.inds = (1+nrow(train2)):nrow(fold2),
```

```
                                        nrow(fold2)),
                par.set = param,
                control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 3
set.seed(125)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb3 <- tuneParams(learn, task = trainTask3,
                resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train3),
                                        test.inds = (1+nrow(train3)):nrow(fold3),
                                        nrow(fold3)),
                par.set = param,
                control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 4
set.seed(126)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb4 <- tuneParams(learn, task = trainTask4,
                resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train4),
                                        test.inds = (1+nrow(train4)):nrow(fold4),
                                        nrow(fold4)),
                par.set = param,
                control = makeTuneControlRandom(maxit = 10L))
########
# Compare result from 6 folds, choose the optimal parameter
# The optimal parameter will be:
```

```
best_xgb4
########
# Assign the optimal hyperparameter to the leaner
learn1 <- setHyperPars(learn, par.vals = best_xgb4$x)
# Fit xgboost model for training data
tr <- mlr::train(learn1, trainTask)
model <- xgb.train(data = dtrain,
                   label = output,
                   objective = "reg:linear",
                   min_child_weight = as.numeric(best_xgb4$x[1]),
                   nrounds = as.numeric(best_xgb4$x[2]),
                   max_depth = as.numeric(best_xgb4$x[3]),
                   eta = as.numeric(best_xgb4$x[4]),
                   subsample = as.numeric(best_xgb4$x[5]),
                   lambda = as.numeric(best_xgb4$x[6]),
                   watchlist = list(train=dtrain),
                   maximize = F , eval_metric = "error")
########################################
# Perform prediction
pred2 <- predict(tr, testTask)
# Calculate accuracy measures
mse_2 <- mse(pred2$data$response, pred2$data$truth)
mae_2 <- mae(pred2$data$response, pred2$data$truth)
rmse_2 <- rmse(pred2$data$response, pred2$data$truth)
mape_2 <- mape(pred2$data$response, pred2$data$truth)
cat("MSE: ", mse_2, "MAE: ", mae_2, "RMSE: ", rmse_2, "MAPE: ", mape_2)
```

```
########################################
# Data frame includes actual and fitted values from GAM and XGBoost
result <- data.frame(test$Market_Index,
                     test_output, pred1, pred2$data$response)
colnames(result) <- c("Market_Index", "Actual", "Fitted_byGAM", "Fitted_byXGBoost")
# Comparison plot between actual values and predicted values based on XGBoost
gamplot <- ggplot(result, aes(x=Actual, y=Fitted_byGAM)) +
  geom_point(color = "pink") +
  labs(title = "GAM") +
  ylab("Fitted values") +
  xlab("Actual values") +
  geom_abline(intercept=0, slope=1, color = "red") + theme(axis.title.x = element_text(size = 20),
                                                           axis.title.y = element_text(size = 20),
                                                           axis.text.x = element_text(size=20, angle=90, hjust=1),
                                                           axis.text.y = element_text(size=22),
                                                           legend.text = element_text(size=20),
                                                           legend.title = element_text(size=20),
                                                           title = element_text(size = 20)
xgplot <- ggplot(result, aes(x=Actual, y=Fitted_byXGBoost)) +
  geom_point(color = "lightblue") +
  labs(title = "XGBoost") +
  ylab("Fitted values") +
  xlab("Actual values") +
  geom_abline(intercept=0, slope=1, color = "red") + theme(axis.title.x = element_text(size = 20),
                                                           axis.title.y = element_text(size = 20),
                                                           axis.text.x = element_text(size=20, angle=90, hjust=1),
```

```
                                    axis.text.y = element_text(size=22),
                                    legend.text = element_text(size=20),
                                    legend.title = element_text(size=20),
                                    title = element_text(size = 20))
# Combine two plots into same page
plot_grid(gamplot, xgplot, labels = "AUTO")


########################################
# 6.1.3. XGBoost Feature Important
########################################
# Find important scores of all features
ximp <- xgb.importance(model = model)
# Create plot of feature importance
xgb.ggplot.importance(importance_matrix = ximp, n_clusters = 1) +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size=20),
        axis.text.y = element_text(size=20),
        legend.text = element_text(size=20),
        legend.title = element_text(size=20),
        title = element_text(size = 20),
        aspect.ratio = 2/3,
        legend.position = "none")
########################################
# Partial dependent plot
# Market index and route
```

```
pd_mr <- generatePartialDependenceData(tr, trainTask, ximp$Feature[1:2])
plotPartialDependence(pd_mr)
# Charterer and owner
pd_co <- generatePartialDependenceData(tr, trainTask, ximp$Feature[c(4,6)])
plotPartialDependence(pd_co)
# Load factor, Lead time and Age
pd_lla <- generatePartialDependenceData(tr, trainTask, ximp$Feature[c(3,5,7)])
plotPartialDependence(pd_lla)
# Interaction between Charterer and Owner
pd_co <- pdp::partial(model, pred.var = c("Owner_encoded", "Charterer_encoded"),
                      train = subset(train, select = -c(Rate_log, Date)),
                      grid.resolution = 10,
                      chull = TRUE,
                      plot = FALSE,
                      .progress = "text")
# 3D
pdp::plotPartial(pd_co, levelplot = FALSE, colorkey = TRUE,
                 screen = list(z = -20, x = -60), zlab = "Rate_log", drape = TRUE)
# 2D
pdp::plotPartial(pd_co, contour = FALSE, shade = TRUE)
########################################
# 6.2. SHAP Values
########################################
shap_values <- shap.values(xgb_model = model, X_train = predictors)
shap_values$mean_shap_score
#SUMMARY PLOT
```

```
shap_long <- shap.prep(xgb_model = model, X_train = predictors)

shap.plot.summary(shap_long, x_bound  = 1.2, dilute = 10)

shap.plot.summary.wrap1(model, X = predictors)

#DEPENDENCE PLOT

fig_list <- lapply(names(shap_values$mean_shap_score)[1:4],

                   shap.plot.dependence, data_long = shap_long, color_feature = 'Market_Index')

grid.arrange(grobs = fig_list, ncol = 2)

fig_list2 <- lapply(names(shap_values$mean_shap_score)[5:7],

                    shap.plot.dependence, data_long = shap_long, color_feature = 'Market_Index')

grid.arrange(grobs = fig_list2, ncol = 2)

#Market_Index dependence plot vs Trend plot of Rate

Market_Index_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Market_Index"]), data_long = shap_long)

#Plot Price over time (Market_Index)

p <- ggplot(train, aes(x=Market_Index, y=Rate_log)) +

  geom_line() +

  xlab("") +

  stat_smooth(

    color = "#FC4E07", fill = "#FC4E07",

    method = "loess"

  )

grid.arrange(Market_Index_ind, p, ncol = 2)

#Route dependence plot

route_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Route_encoded"]), data_long = shap_long)

route_ind

#Charterer dependence plot
```

```
charterer_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Charterer_encoded"]), data_long = shap_long)

charterer_ind

#Owner dependence plot

owner_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Owner_encoded"]), data_long = shap_long)

owner_ind

#Lead time dependence plot

lt_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Lead_time"]), data_long = shap_long)

lt_ind

#Load factor dependence plot

lf_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Load_Factor"]), data_long = shap_long)

lf_ind

#Age dependence plot

age_ind <- shap.plot.dependence(x=names(shap_values$mean_shap_score["Age"]), data_long = shap_long)

age_ind

grid.arrange(charterer_ind, owner_ind , ncol=2)

########################

# LOCAL EXPLANATION

#On train set

shap_data <- shap_values$shap_score

shap_data[, BIAS := shap_values$BIAS0]

pred_mod <- predict(model, predictors)

shap_data[, `:=`(Row_Sum = round(rowSums(shap_data),6), Pred_Mod = round(pred_mod,6))]

#Measure SHAP values on test set
```

```
shap_values_test <- shap.values(xgb_model = model, X_train = test[,-8])

shap_data_test <- shap_values_test$shap_score

#Measure base value

shap_data_test[, BIAS := shap_values_test$BIAS0]

shap_data_test[, `:=`(rowSum = round(rowSums(shap_data_test),6), pred_mod =
round(pred2$data$response,6))]

names(shap_data_test) <- c("Charterer_SV", "Route_SV", "Owner_SV", "Lead_time_SV",

                "Age_SV", "Load_Factor_SV", "Market_Index_SV", "BIAS", "Row_Sum",
"Pred_Mod")

test_temp2 <- cbind(test_temp, shap_data_test)

test_temp3 <- test_temp2 %>%

  select(Charterer, Charterer_SV, Route, Route_encoded, Route_SV, Owner, Owner_SV,

        Lead_time, Lead_time_SV, Age, Age_SV, Load_Factor, Load_Factor_SV,

        Market_Index, Market_Index_SV, BIAS, Row_Sum, Pred_Mod, Rate)

##############################################

#INTERACTION EFFECT

# vs Time trend (i.e., Market_Index)

shap_int <- predict(model, predictors, predinteraction = TRUE)

charterer_t <- shap.plot.dependence(data_long = shap_long,

                                    data_int = shap_int,

                                    x= "Market_Index", y = "Charterer_encoded",

                                    color_feature = "Charterer_encoded")

route_t <- shap.plot.dependence(data_long = shap_long,

                                data_int = shap_int,

                                x= "Market_Index", y = "Route_encoded",

                                color_feature = "Route_encoded")

owner_t <- shap.plot.dependence(data_long = shap_long,
```

```
                        data_int = shap_int,
                        x= "Market_Index", y = "Owner_encoded",
                        color_feature = "Owner_encoded")
leadtime_t <- shap.plot.dependence(data_long = shap_long,
                          data_int = shap_int,
                          x= "Market_Index", y = "Lead_time",
                          color_feature = "Lead_time", dilute = 10)
age_t <- shap.plot.dependence(data_long = shap_long,
                       data_int = shap_int,
                       x= "Market_Index", y = "Age",
                       color_feature = "Age")
lf_t <- shap.plot.dependence(data_long = shap_long,
                      data_int = shap_int,
                      x= "Market_Index", y = "Load_Factor",
                      color_feature = "Load_Factor")
grid.arrange(charterer_t, route_t, ncol=2)
grid.arrange(owner_t,leadtime_t,  ncol=2)
grid.arrange(age_t, lf_t, ncol=2)
#vs Charterer
Market_Index_c <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Charterer_encoded", y = "Market_Index",
                        color_feature = "Market_Index")
route_c <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Charterer_encoded", y = "Route_encoded",
```

```
                        color_feature = "Route_encoded", dilute =8)
owner_c <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Charterer_encoded", y = "Owner_encoded",
                        color_feature = "Owner_encoded", dilute = 8)
leadtime_c <- shap.plot.dependence(data_long = shap_long,
                          data_int = shap_int,
                          x= "Charterer_encoded", y = "Lead_time",
                          color_feature = "Lead_time")
age_c <- shap.plot.dependence(data_long = shap_long,
                       data_int = shap_int,
                       x= "Charterer_encoded", y = "Age",
                       color_feature = "Age")
lf_c <- shap.plot.dependence(data_long = shap_long,
                      data_int = shap_int,
                      x= "Charterer_encoded", y = "Load_Factor",
                      color_feature = "Load_Factor")
grid.arrange(route_c, owner_c, ncol=2)
grid.arrange(leadtime_c2, age_c, ncol=2)
grid.arrange(lf_c, Market_Index_c, ncol=2)
#vs Route
Market_Index_r <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Route_encoded", y = "Market_Index",
                        color_feature = "Market_Index")
```

```
charterer_r <- shap.plot.dependence(data_long = shap_long,
                                    data_int = shap_int,
                                    x= "Route_encoded", y = "Charterer_encoded",
                                    color_feature = "Charterer_encoded")
owner_r <- shap.plot.dependence(data_long = shap_long,
                                data_int = shap_int,
                                x= "Route_encoded", y = "Owner_encoded",
                                color_feature = "Owner_encoded")
leadtime_r <- shap.plot.dependence(data_long = shap_long,
                                   data_int = shap_int,
                                   x= "Route_encoded", y = "Lead_time",
                                   color_feature = "Lead_time")
age_r <- shap.plot.dependence(data_long = shap_long,
                              data_int = shap_int,
                              x= "Route_encoded", y = "Age",
                              color_feature = "Age", dilute = 8)
lf_r <- shap.plot.dependence(data_long = shap_long,
                             data_int = shap_int,
                             x= "Route_encoded", y = "Load_Factor",
                             color_feature = "Load_Factor")
grid.arrange(charterer_r, owner_r, ncol=2)
grid.arrange(leadtime_r, age_r, ncol=2)
grid.arrange(lf_r, Market_Index_r, ncol=2)
# vs Owner
Market_Index_o <- shap.plot.dependence(data_long = shap_long,
                                       data_int = shap_int,
```

```
                        x= "Owner_encoded", y = "Market_Index",
                        color_feature = "Market_Index")
route_o <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Owner_encoded", y = "Route_encoded",
                        color_feature = "Route_encoded")
charterer_o <- shap.plot.dependence(data_long = shap_long,
                          data_int = shap_int,
                          x= "Owner_encoded", y = "Charterer_encoded",
                          color_feature = "Charterer_encoded")
leadtime_o <- shap.plot.dependence(data_long = shap_long,
                          data_int = shap_int,
                          x= "Owner_encoded", y = "Lead_time",
                          color_feature = "Lead_time")
age_o <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Owner_encoded", y = "Age",
                        color_feature = "Age")
lf_o <- shap.plot.dependence(data_long = shap_long,
                       data_int = shap_int,
                       x= "Owner_encoded", y = "Load_Factor",
                       color_feature = "Load_Factor")
grid.arrange(route_o, charterer_o, ncol=2)
grid.arrange(leadtime_o, age_o, ncol=2)
grid.arrange(lf_o, Market_Index_o, ncol=2)
#vs Lead time
```

```
Market_Index_lt <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Lead_time", y = "Market_Index",
                        color_feature = "Market_Index")
route_lt <- shap.plot.dependence(data_long = shap_long,
                         data_int = shap_int,
                         x= "Lead_time", y = "Route_encoded",
                         color_feature = "Route_encoded")
charterer_lt <- shap.plot.dependence(data_long = shap_long,
                           data_int = shap_int,
                           x= "Lead_time", y = "Charterer_encoded",
                           color_feature = "Charterer_encoded")
owner_lt <- shap.plot.dependence(data_long = shap_long,
                         data_int = shap_int,
                         x= "Lead_time", y = "Owner_encoded",
                         color_feature = "Owner_encoded")
age_lt <- shap.plot.dependence(data_long = shap_long,
                       data_int = shap_int,
                       x= "Lead_time", y = "Age",
                       color_feature = "Age")
lf_lt <- shap.plot.dependence(data_long = shap_long,
                      data_int = shap_int,
                      x= "Lead_time", y = "Load_Factor",
                      color_feature = "Load_Factor")
grid.arrange(route_lt, charterer_lt, ncol=2)
grid.arrange(owner_lt, age_lt, ncol=2)
```

```
grid.arrange(lf_lt, Market_Index_lt, ncol=2)
#vs Age
Market_Index_a <- shap.plot.dependence(data_long = shap_long,
                        data_int = shap_int,
                        x= "Age", y = "Market_Index",
                        color_feature = "Market_Index")
route_a <- shap.plot.dependence(data_long = shap_long,
                         data_int = shap_int,
                         x= "Age", y = "Route_encoded",
                         color_feature = "Route_encoded", dilute=8)
charterer_a <- shap.plot.dependence(data_long = shap_long,
                            data_int = shap_int,
                            x= "Age", y = "Charterer_encoded",
                            color_feature = "Charterer_encoded")
owner_a <- shap.plot.dependence(data_long = shap_long,
                         data_int = shap_int,
                         x= "Age", y = "Owner_encoded",
                         color_feature = "Owner_encoded")
leadtime_a <- shap.plot.dependence(data_long = shap_long,
                           data_int = shap_int,
                           x= "Age", y = "Lead_time",
                           color_feature = "Lead_time")
lf_a <- shap.plot.dependence(data_long = shap_long,
                       data_int = shap_int,
                       x= "Age", y = "Load_Factor",
                       color_feature = "Load_Factor")
```

```
grid.arrange(route_a, charterer_a, ncol=2)
grid.arrange(owner_a, leadtime_a, ncol=2)
grid.arrange(lf_a, Market_Index_a, ncol=2)
#vs Load Factor
Market_Index_lf <- shap.plot.dependence(data_long = shap_long,
                          data_int = shap_int,
                          x= "Load_Factor", y = "Market_Index",
                          color_feature = "Market_Index")
route_lf <- shap.plot.dependence(data_long = shap_long,
                           data_int = shap_int,
                           x= "Load_Factor", y = "Route_encoded",
                           color_feature = "Route_encoded")
charterer_lf <- shap.plot.dependence(data_long = shap_long,
                             data_int = shap_int,
                             x= "Load_Factor", y = "Charterer_encoded",
                             color_feature = "Charterer_encoded")

owner_lf <- shap.plot.dependence(data_long = shap_long,
                           data_int = shap_int,
                           x= "Load_Factor", y = "Owner_encoded",
                           color_feature = "Owner_encoded")
leadtime_lf <- shap.plot.dependence(data_long = shap_long,
                             data_int = shap_int,
                             x= "Load_Factor", y = "Lead_time",
                             color_feature = "Lead_time")
age_lf <- shap.plot.dependence(data_long = shap_long,
```

```
                    data_int = shap_int,
                    x= "Load_Factor", y = "Age",
                    color_feature = "Age")
grid.arrange(route_lf, charterer_lf, ncol=2)
grid.arrange(owner_lf, leadtime_lf, ncol=2)
grid.arrange(age_lf, Market_Index_lf, ncol=2)
#Generate tables of SHAP values
names(shap_data) <- c("Charterer_SV", "Route_SV", "Owner_SV", "Lead_time_SV",
                      "Age_SV",  "Load_Factor_SV",  "Market_Index_SV",  "BIAS",  "Row_Sum",
"Pred_Mod")
train_temp2 <- cbind(train_temp, shap_data)
train_temp3 <- train_temp2 %>%
  select(Charterer, Charterer_SV, Route, Route_encoded, Route_SV, Owner, Owner_SV,
         Lead_time, Lead_time_SV, Age, Age_SV, Load_Factor, Load_Factor_SV,
         Market_Index, Market_Index_SV, BIAS, Row_Sum, Pred_Mod, Rate)
#SHAP values of top 10 charterers
route_SV <- train_temp3 %>%
  group_by(Route) %>%
  summarise(Route_encoded = mean(Route_encoded),
    Route_SV = mean(Route_SV),
    Fixtures = n()) %>%
  arrange(desc(Route_SV))
#SHAP values of top 10 charterers
charterer_SV <- train_temp3 %>%
  group_by(Charterer) %>%
  summarise(Charterer_SV = mean(Charterer_SV),
            Fixtures = n()) %>%
```

```
  arrange(desc(Fixtures)) %>%
  top_n(n=10, Fixtures)
#SHAP values of top 10 owners
owner_SV <- train_temp3 %>%
  group_by(Owner) %>%
  summarise(Owner_SV = mean(Owner_SV),
            Fixtures = n()) %>%
  arrange(desc(Fixtures)) %>%
  top_n(n=10, Fixtures)


#################################################################
# Appendix
# A6. Fixed Effect Linear Regression
#################################################################
# Standardize the numeric variable for linear regression
# Creating Age Squared variable
df_lr <- df %>%
  mutate(Age_sqr = Age^2)
# Numeric variables
DFnumeric <- df_lr[,c("Lead_time","Age", "Load_Factor", "Age_sqr")]
# Standardize the data
PreNum <- preProcess(DFnumeric, method=c("center", "scale"))
DFnorm <- predict(PreNum, DFnumeric)
summary(DFnorm)
df_lr <- cbind(df_lr[,c("Charterer", "Route","Owner", "Market_Index", "Rate_log")], DFnorm)
##############################
```

```
set.seed(6543)

# Fit model

lr1 <- feols(Rate_log ~ Lead_time + Age + Age_sqr + Load_Factor + Route | Market_Index,

             data = df_lr) #Time fixed effect

lr2 <- feols(Rate_log ~ Lead_time + Age + Age_sqr + Load_Factor + Route | Charterer + Owner,

             data = df_lr) #Two-way fixed effect

lr3 <- feols(Rate_log ~ Lead_time + Age + Age_sqr + Load_Factor + Route | Market_Index +
Charterer + Owner,

             data = df_lr) #Three way fixed effect

# Summary of fixed effects OLS estimation

summary(lr3, cluster = "Route")

etable(lr1,lr2,lr3, cluster = "Route")

# Extracting the fixed-effects coefficients

fe <- fixef(lr3)

summary(fe)

fe$Market_Index

fe$Charterer

fe$Owner

#Plot the most notable fixed effects

plot(fe)


#####################################################################

# A7. GAM for full data

# Target Encoding

#Apply on tuning data

df[["Route_encoded"]] <- encode_target(df[["Route"]], df[["Rate_log"]])

df[["Charterer_encoded"]] <- encode_target(df[["Charterer"]], df[["Rate_log"]])
```

```
df[["Owner_encoded"]] <- encode_target(df[["Owner"]], df[["Rate_log"]])

# Get the final data

df_xg <- df %>%

  select(Date, Charterer_encoded, Route_encoded, Owner_encoded, Lead_time, Age, Load_Factor, Market_Index, Rate_log)

########################

# Fit model

set.seed(5678)

gam1<-
gam(Rate_log~s(Lead_time)+s(Age)+s(Load_Factor)+s(Market_Index)+s(Charterer_encoded)+s(Owner_encoded)

          +s(Route_encoded),

          data = df_xg[,-1], method="REML", select = TRUE)

# Summary of GAM results

summary(gam1)

# Plot partial dependence plots

plot(gam1, pages = 4, residuals=F, pch=19, cex=0.25,

     scheme=1, col='#FF8000', shade=T,shade.col='gray90')

# Check GAM residuals

mar <- par(mfrow = c(2, 2))

gam.check(gam1, old.style=FALSE,

          type=c("deviance","pearson","response"))

par(mar)


################

# A8. XGBoost for full data

# Time series cross validation

tscv <- df_xg %>%
```

```
  time_series_cv(
    date_var    = Date,
    initial     = "3 years",
    assess      = "1 year",
    skip        = "1 year",
    slice_limit = 10
  )
tscv %>%
  plot_time_series_cv_plan(Date, Rate_log, .interactive = FALSE)
# Divide data into 6 folds:
# Fold 1:
train1 <- df_xg[c(tscv$splits[[1]]$in_id[1]:tail(tscv$splits[[1]]$in_id, n=1)),-1]
vali1 <- df_xg[c(tscv$splits[[1]]$out_id[1]:tail(tscv$splits[[1]]$out_id, n=1)),-1]
fold1 <- rbind(train1, vali1)
# Fold 2:
train2 <- df_xg[c(tscv$splits[[2]]$in_id[1]:tail(tscv$splits[[2]]$in_id, n=1)),-1]
vali2 <- df_xg[c(tscv$splits[[2]]$out_id[1]:tail(tscv$splits[[2]]$out_id, n=1)),-1]
fold2 <- rbind(train2, vali2)
# Fold 3:
train3 <- df_xg[c(tscv$splits[[3]]$in_id[1]:tail(tscv$splits[[3]]$in_id, n=1)),-1]
vali3 <- df_xg[c(tscv$splits[[3]]$out_id[1]:tail(tscv$splits[[3]]$out_id, n=1)),-1]
fold3 <- rbind(train3, vali3)
# Fold 4:
train4 <- df_xg[c(tscv$splits[[4]]$in_id[1]:tail(tscv$splits[[4]]$in_id, n=1)),-1]
vali4 <- df_xg[c(tscv$splits[[4]]$out_id[1]:tail(tscv$splits[[4]]$out_id, n=1)),-1]
fold4 <- rbind(train4, vali4)
```

```
# Fold 5:
train5 <- df_xg[c(tscv$splits[[5]]$in_id[1]:tail(tscv$splits[[5]]$in_id, n=1)),-1]
vali5 <- df_xg[c(tscv$splits[[5]]$out_id[1]:tail(tscv$splits[[5]]$out_id, n=1)),-1]
fold5 <- rbind(train5, vali5)
# Fold 6:
train6 <- df_xg[c(tscv$splits[[6]]$in_id[1]:tail(tscv$splits[[6]]$in_id, n=1)),-1]
vali6 <- df_xg[c(tscv$splits[[6]]$out_id[1]:tail(tscv$splits[[6]]$out_id, n=1)),-1]
fold6 <- rbind(train6, vali6)
########################
# Prepare for fitting XGBoost model
# The predictor variables
predictors <- df_xg %>%
  select(-c(Rate_log, Date)) %>%
  as.matrix()
# The response variable
output <- df_xg$Rate_log
# Constract xgb.DMatrix object for XGBoost
dtrain <- xgb.DMatrix(data = predictors, label = output)
# Change column names of training, validation and test data
colnames(fold1) <- make.names(colnames(fold1),unique = T)
colnames(fold2) <- make.names(colnames(fold2),unique = T)
colnames(fold3) <- make.names(colnames(fold3),unique = T)
colnames(fold4) <- make.names(colnames(fold4),unique = T)
colnames(fold5) <- make.names(colnames(fold5),unique = T)
colnames(fold6) <- make.names(colnames(fold6),unique = T)
########################
```

```
# XGBoost
# Tuning Hyperparameters
# create mlr task for XGBoost
trainTask <- makeRegrTask(data = df_xg[,-1], target = "Rate_log")
trainTask1 <- makeRegrTask(data = fold1, target = "Rate_log")
trainTask2 <- makeRegrTask(data = fold2, target = "Rate_log")
trainTask3 <- makeRegrTask(data = fold3, target = "Rate_log")
trainTask4 <- makeRegrTask(data = fold4, target = "Rate_log")
trainTask5 <- makeRegrTask(data = fold5, target = "Rate_log")
trainTask6 <- makeRegrTask(data = fold6, target = "Rate_log")
# Fold 1
set.seed(123)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb1 <- tuneParams(learn, task = trainTask1,
                        resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train1),
                                                              test.inds = (1+nrow(train1)):nrow(fold1),
                                                              nrow(fold1)),
                        par.set = param,
                        control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 2
set.seed(124)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb2 <- tuneParams(learn, task = trainTask2,
                        resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train2),
                                                              test.inds = (1+nrow(train2)):nrow(fold2),
```

```
                                        nrow(fold2)),
                par.set = param,
                control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 3
set.seed(125)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb3 <- tuneParams(learn, task = trainTask3,
                resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train3),
                                        test.inds = (1+nrow(train3)):nrow(fold3),
                                        nrow(fold3)),
                par.set = param,
                control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 4
set.seed(126)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb4 <- tuneParams(learn, task = trainTask4,
                resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train4),
                                        test.inds = (1+nrow(train4)):nrow(fold4),
                                        nrow(fold4)),
                par.set = param,
                control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 5
set.seed(127)
```

```
# Run base learner for the training Task and randomly search throuhgh 10 iterations
best_xgb5 <- tuneParams(learn, task = trainTask5,
                        resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train5),
                                                              test.inds = (1+nrow(train5)):nrow(fold5),
                                                              nrow(fold5)),
                        par.set = param,
                        control = makeTuneControlRandom(maxit = 10L))
#########
# Fold 6
set.seed(128)
# Run base learner for the training Task and randomly search through 10 iterations
best_xgb6 <- tuneParams(learn, task = trainTask6,
                        resampling = makeFixedHoldoutInstance(train.inds = 1:nrow(train6),
                                                              test.inds = (1+nrow(train6)):nrow(fold6),
                                                              nrow(fold6)),
                        par.set = param,
                        control = makeTuneControlRandom(maxit = 10L))
########
# Compare result from 6 folds, choose the optimal parameter
# The optimal parameter will be:
best_xgb3


########
# Assign the optimal hyperparameter to the leaner
learn1 <- setHyperPars(learn, par.vals = best_xgb3$x)
# Fit xgboost model for training data
```

```
tr <- mlr::train(learn1, trainTask)
model <- xgb.train(data = dtrain,
                   label = output,
                   objective = "reg:linear",
                   min_child_weight = as.numeric(best_xgb3$x[1]),
                   nrounds = as.numeric(best_xgb3$x[2]),
                   max_depth = as.numeric(best_xgb3$x[3]),
                   eta = as.numeric(best_xgb3$x[4]),
                   subsample = as.numeric(best_xgb3$x[5]),
                   lambda = as.numeric(best_xgb3$x[6]),
                   watchlist = list(train=dtrain),
                   maximize = F , eval_metric = "error")


########################################
# XGBoost Feature Important
########################################
# Find important scores of all features
ximp <- xgb.importance(model = model)
# Create plot of feature importance
xgb.ggplot.importance(importance_matrix = ximp, n_clusters = 1) +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text.x = element_text(size=20),
        axis.text.y = element_text(size=20),
        legend.text = element_text(size=20),
        legend.title = element_text(size=20),
```

```
      title = element_text(size = 20),
      aspect.ratio = 2/3,
      legend.position = "none")


########################################
# Partial dependence plot
pd <- generatePartialDependenceData(tr, trainTask, ximp$Feature)
plotPartialDependence(pd)
```